1994-03

Corrective software management: the success of the EPLRS program

Cummings, Terrence

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

CORRECTIVE SOFTWARE MANAGEMENT:
THE SUCCESS OF THE EPLRS PROGRAM

by

Terrence Cummings

March 1994

Principal Advisor: Martin J. McCaffrey

Associate Advisor: Tarek Abdel-Hamid

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

1. AGENCY USE ONLY	2. REPORT DATE March 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Corrective Software Management: The Success of the EPLRS Program

5. FUNDING NUMBERS

6. AUTHOR(S) Cummings, Terrence

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT
Software development in weapon systems is extremely challenging and has become a significant source of problems in DOD programs. The purpose of this thesis is to identify and document strategies and techniques for program offices to use in managing these software problems. This thesis documents the success of the Enhanced Position Location Reporting System in applying corrective software management actions to specific problems encountered. Lessons learned have been drawn from the analysis and generalized for application to other DOD programs. The principal finding is that an effective software corrective action plan requires a focused effort devoted to identifying and correcting all software deficiencies. This is accomplished before further developmental work requiring software is attempted. The thesis concludes that an astute program office should be prepared to implement and manage this type of software corrective action plan. Two primary recommendations are for the development of a DOD policy on the management of software corrective action and the development of a DOD model for software corrective action by program offices.

14. SUBJECT TERMS Mission Critical Computer Resources (MCCR), Software Development, Enhanced Position Location Reporting System (EPLRS), Software Engineering Institute (SEI), Army Data Distribution System (ADDS), Technical Test, Prototype Qualification Test-Contractor (PQT-C)

15. NUMBER OF PAGES
124

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

Corrective Software Management:
The Success of the EPLRS Program

by

Terrence Cummings
Captain, United States Army
BS., United States Military Academy, 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
March 1994

Author: [redacted]

Terrence Cummings

Approved by: [redacted]

Martin J. McCaffrey, Principal Advisor

[redacted]

Tarek Abdel-Hamid, Associate Advisor

[redacted]

David R. Whipple, Chairman
Department of Systems Management

ABSTRACT

Software development in weapon systems is extremely challenging and has become a significant source of problems in DOD programs. The purpose of this thesis is to identify and document strategies and techniques for program offices to use in managing these software problems. This thesis documents the success of the Enhanced Position Location Reporting System in applying corrective software management actions to specific problems encountered. Lessons learned have been drawn from the analysis and generalized for application to other DOD programs. The principal finding is that an effective software corrective action plan requires a focused effort devoted to identifying and correcting all deficiencies in the software. This is accomplished before further system development work requiring software is attempted. The thesis concludes that an astute program office should be prepared to implement and manage this type of software corrective action plan. Two primary recommendations are for the development of a DOD policy on the management of software corrective action and the development of a DOD model for software corrective action by program offices.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. GENERAL

Over the past twenty years, the technological advancement of Department of Defense (DOD) weapon systems has relied heavily on increasingly complex computer software. Management of the design and development of this software is extremely challenging and has become a major source of problems in the systems acquisition field. Program offices are commonly faced with problems in software acquisition that call for some type of corrective action. To date, little documentation exists that details successful strategies and techniques available to a program office for use in correcting problems encountered during software development. This study will document the success of one program office in applying corrective software management actions to specific problems encountered. Successful, general concepts and strategies for corrective software management can be captured from this case for use in future work.

B. OBJECTIVES

This thesis will examine corrective software management in the Enhanced Position Location and Reporting System (EPLRS) acquisition program of the Department of the Army. The specific corrective actions in software management for this program will be examined and identified. These actions will then be analyzed for their general application to software management problems during acquisition. The overall objective of this thesis is to document the knowledge that will assist in developing successful strategies for dealing with software management problems.

C. SCOPE

This thesis will focus on the corrective action plan developed during the acquisition of the Army's Enhanced Position Location Reporting System (EPLRS). The central concern will be the challenges faced by the product managers after poor technical test results in 1988 and 1989. The events leading up to the poor test results are not of direct concern in this thesis and will only be covered to an extent necessary to illustrate the need for corrective action.

The perspective of this research and analysis will be managerial, not technical. Obviously, managers in this area must be adequately familiar with the technical aspects of their program to be successful. With this in mind, technical aspects of the EPLRS program will be presented as they apply to managerial decisions.

This research focused on mission critical software. Mission critical software is a component of Mission Critical Computer Resources (MCCR), which refers to the totality of computer hardware and software integral to a weapon system. Section 908 of Public Law 97-86, also known as the Warner-Nunn Amendment, defined MCCR as those computer resources that perform the following functions, operations, or use:
[Ref. 2:p. 4-3]

1. Involves intelligence activities;
2. Involves cryptoanalytic activities related to national security;

3. Involves the command and control of military forces;

4. Involves equipment that is an integral part of a weapon system;

5. Is critical to the direct fulfillment of military or intelligence missions.

The term 'MCCR system', when used in this document, refers to defense systems as defined above. Mission critical software excludes all administrative type computer software programs.

D. METHODOLOGY

This thesis was conducted as a case study. First, an historical overview of weapon system software acquisition was conducted. This was accomplished by reviewing material from professional manuals, periodicals, and previous theses. This provided the necessary background knowledge from an historical perspective from which to begin the case analysis. Analysis of the case required information on the status of the program at various phases of its development, identity and roles of key personnel, decisions and subsequent actions that affected the program, and an evaluation of these actions. This information was gathered through personal interviews with government personnel working directly and indirectly with the EPLRS program, and contractor personnel working on the EPLRS program for Hughes Aircraft Company. Additionally, pertinent documents from the EPLRS program office and Hughes Aircraft Company, and related periodicals and government reports were reviewed.

E. ORGANIZATION

Chapter II establishes the background for the study by overviewing the role of software in DOD acquisition programs. The chapter illustrates the history of software problems in DOD acquisition programs and will define the term "corrective software management."

Chapter III describes the form of case study methodology used in this thesis. The chapter explains the applicability of the case study strategy to the thesis and illustrates the case study research design used.

Chapter IV introduces the reader to the EPLRS system and the agencies responsible for it. It briefly details the history and acquisition strategy of the program. The chapter closes by describing the program technical testing that prompted corrective action for the system.

Chapter V describes the corrective action used by the Government and the contractor to correct system deficiencies identified during technical testing. It presents the initial Government guidance, a review of the corrective action plan, and a discussion of the key actions taken by the Government and the contractor. The chapter also discusses the results of the corrective action.

Chapter VI analyzes the events and decisions critical to the corrective action plan of the EPLRS program. Additionally, it will discuss the applicability of this case to other DOD programs. The chapter then presents a list of lessons learned from the EPLRS case that can be generalized to other DOD acquisition programs.

Chapter VII presents a conclusion of the analysis. The chapter also presents recommendations and answers to the research questions. The chapter will close with topics for further study and a final note on the issue.

II. BACKGROUND

A. INTRODUCTION

Software development as a recognized activity is relatively young. Its significant history with the DOD dates back less than 45 years when weapon systems used analog computers to automate basic functions. Software engineering is even younger, highlighted by the fact that computer science departments were just being established as separate entities in colleges in the late 1960's. It's defined as the technological and managerial discipline concerned with systematic production and maintenance of software products that are developed and modified on time and within cost estimates [Ref. 1]. The term "Software Engineering" was first used in 1968. [Ref. 2] As an engineering discipline it is still in its infancy. This fact is illustrated by the scarcity and immaturity of the practices, procedures, and tools used in the software engineering field as compared to other engineering disciplines, many of which have evolved over a hundred years or more.

Nonetheless, software development and engineering has progressed steadily during its lifetime. The introduction of digital systems in the mid 1950's greatly expanded the application of computers and software in military systems. Software's flexibility and cost effectiveness for change, as compared to computer hardware, has made it a desirable element in the development of DOD weapon systems. As a result, the demand for military software has increased rapidly during the last three decades. All of the services have contributed to this demand by developing increased weapon system capability through the use of software. Examples of this wide variety of weapon systems include the Army's M1A2 main battle tank, the Navy's Trident II missile, and the Air Force's B-1B bomber. Each of these systems is software intensive. Today all major weapon systems are dependent upon software in some way.

B. THE ROLE OF SOFTWARE IN DOD WEAPON SYSTEMS

1. General

In 1987, the "Report of the Defense Science Board Task Force on Military Software" described the role of military software in this way:

> Software plays a major role in today's weapon systems. The "smarts" of smart weapons are provided by software. Software is crucial to intelligence, communications, command, and control.[...]Software provides a major component of U.S. war-fighting capability.[Ref. 3]

The use of embedded software provides the ability to change or increase the functionality and capabilities of a weapon system, often with little or no effect on the hardware characteristics. Software performs many of the critical functions in key weapon systems that cannot be performed by hardware alone. In essence, our key weapon systems today are completely dependent upon software to function properly.

2. Software Size, Growth, and Complexity

An objective of U.S. National Defense Strategy is to maintain technological superiority in weapon systems. [Ref. 4] The "high-tech" weapons that have evolved under this strategy during the last three decades have seen an exponential growth in software costs as a percentage of total computer resource cost. As shown in Figure 1, below, the ratio of computer resource costs in major weapon systems changed from 80% hardware and 20% software in 1960 to 20% hardware and 80% software in 1980 [Ref. 5].

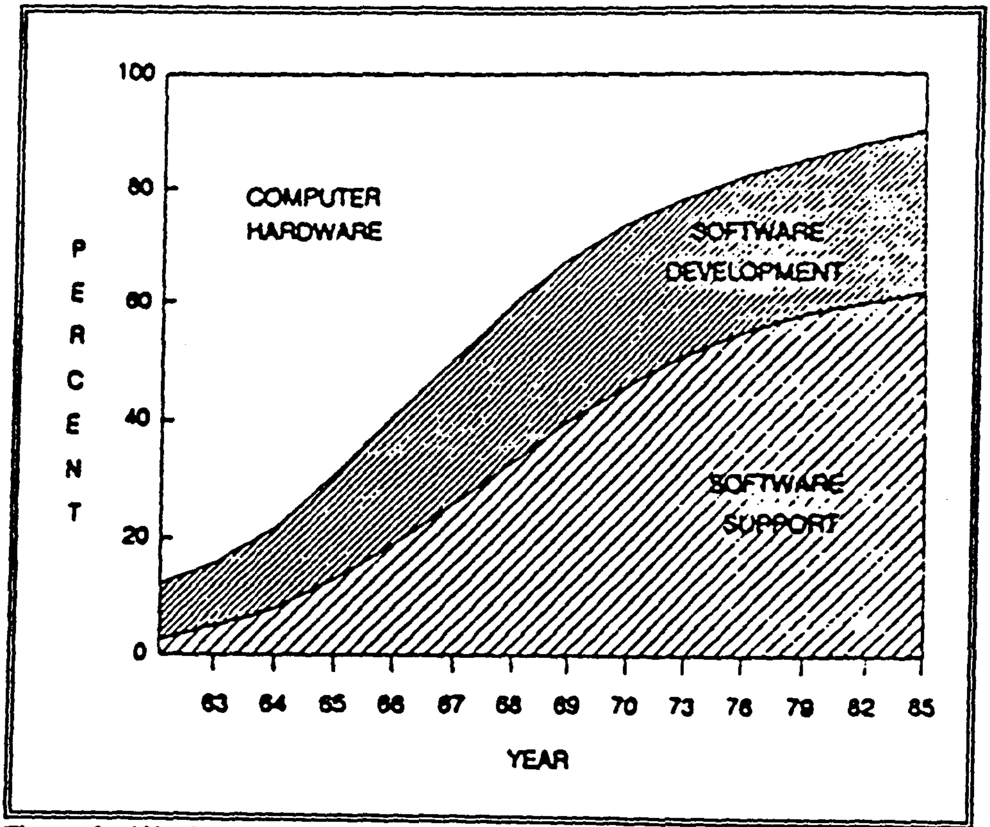

Figure 1. Life Cycle Cost Trends

This growth in software cost has primarily been a result of the growth in the volume and complexity of software demanded by DOD. As weapon systems have become more capable and complex over the years, the software associated with them has grown dramatically. For example, the F-4 aircraft of the Vietnam war era had practically no software. Today's F-14D aircraft currently relies on over 1 million source lines of code (SLOC) to perform its mission. And, in the near future, estimates predict that the Advanced Tactical Fighter (ATF) will require approximately 7 million SLOC to operate.[Ref. 6] This represents an increase not only in volume, but also in software complexity. This complex software costs more to develop and support after fielding. Similar increases in software volume and complexity are evident in every category of system that depends upon Mission Critical Computer Resources (MCCR).

The total volume of DOD software resulting from this demand is staggering. A technical report by the SEI estimated the DOD demand for Ada language alone in 1989 was over 40 million lines of code requiring a rough estimate of over 9,000 person-years of programming effort based on moderate code difficulty.[Ref. 7] This work estimated the number of lines of Ada programming code planned, in full scale development, and in the post deployment software support (PDSS) stage. Both figures are considered underestimates. When one considers the MCCR programs using languages other than Ada and the trend towards increasing software complexity and volume, the current amount of weapon system software is astonishing.

3. Software Costs

Producing this massive amount of weapon system software comes at no small cost to the Government. While cost data on DOD programs have been poorly tracked in the past, 1992 estimates of total software expenditures ranged from $24 billion to $32 billion. This was approximately 8-11% of the DOD budget for that year. In the next

15 years it is estimated that software may increase to an annual cost of $50 billion and account for up to 20% of the DOD budget. [Ref. 9]

To the program office developing or producing a MCCR system, the cost of software has become one of the system's most expensive elements. The software developmental costs for software intensive systems can result in large portions of the programs budget. Table 1 provides some examples of the software developmental cost and its percentage of the total developmental cost of selected DOD MCCR systems [Ref. 8].

TABLE 1. SOFTWARE DEVELOPMENT COSTS

SERVICE	PROGRAM	SOFTWARE DEVELOPMENT COST	% TOTAL DEVELOPMENT COST
Air Force	Adv Tactical Fighter	$1 Billion	13%
Air Force	B-1B Bomber	$726 Million	19%
Army	LHX Helicopter	$115 Million	3%
Navy	SSN-21 Submarine	$450 Million	13%
Navy	Trident II Missile	$280 Million	9%

With respect to volume, complexity, and cost, as well as functionality, software is a critical component in all MCCR systems. Without exception, performance of today's complex weapon systems is dependant upon the associated software. Software has grown into a multi-billion dollar facet of the defense procurement process and it clearly plays a critical role in DOD's weapon systems.

C. SOFTWARE DEVELOPMENT PROBLEMS IN DOD

1. General

As the role of software in DOD's mission critical systems has grown, so, too, has the significance of software development problems. Software developmental problems today have been referred to by some as a "software crisis."[Ref. 9] Air Force General Bernard Rogers has characterized software as the Achilles' heel of weapon system development [Ref. 10]. By any account, software development has become one of the most significant sources of trouble to DOD weapon development programs. The Defense Systems Management College's *Mission Critical Computer Resources Management Guide* describes the impact of software development problems on military weapon systems in this way:

> Most systems are delivered late, have cost overruns, rarely meet performance requirements upon initial delivery and are often ridiculously expensive to maintain. It would be unfair to blame all of these unpleasant facts just on digital systems and software, but it is generally recognized that software is a major contributor, and often the only contributor, to these problems.[Ref. 6]

2. Major Contributors

There is a wide variety of software development problems that plague DOD programs. Figure 2 depicts the classic problems that contribute to the challenge facing program offices [Ref. 2:p. 9-1]. Problems from this list have surfaced over the past several years in the development of numerous weapon systems causing problems with cost, schedule, and performance.

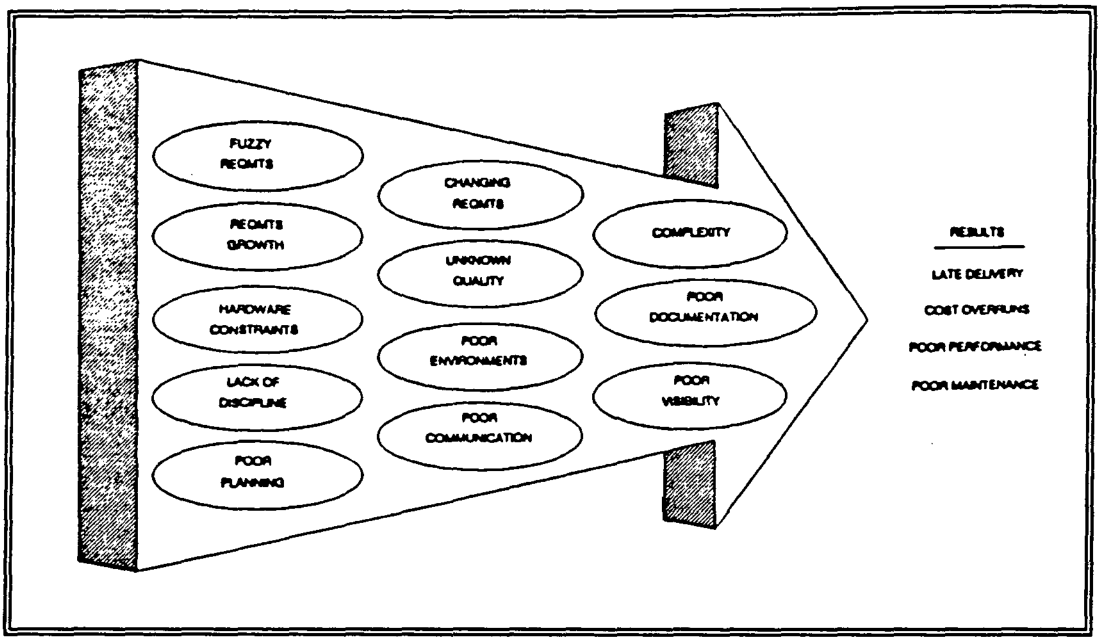

Figure 2. Classic Software Development Problems

A 1992 General Accounting Office (GAO) report on mission critical software challenges grouped the majority of these software development problems into three categories: (1) management, (2) requirements definition, and (3) testing. Management problems are those that program managers have direct control over, such as lack of management oversight, unsound development approaches, and poor management decisions. Requirements definition problems are those that involve the clear depiction of what a system is supposed to do. These problems include poorly defined requirements, requirements changes and a system's inability to adapt to changing requirements. Finally, testing problems are those that involve flawed approaches to testing the system, such as omitting system level integration testing or testing conducted in inaccurate environments or using inaccurate models. [Ref. 12]

While not all inclusive, these categories of problems contain the major contributors to significant cost increases, schedule delays, and performance shortfalls that have troubled DOD programs.

From another perspective, software development problems can be categorized as managerial problems or technical problems. This perspective looks more at the source or cause of the problems listed above. According to the *Report of the Defense Science Board Task Force on Military Software,* the major problems with military software development are management problems, not technical problems. The Board considered these problems anchored in the attitudes, policies, and practices of DOD's software acquisition process. [Ref. 3:p. 7]

3. Effects on the System

The contributing software development problems discussed above have had a negative impact on the majority of MCCR programs during the development and production phase of their system's life-cycle. Under the Secretary of Defense's office, an official from the Office of the Deputy Director for Research and Engineering (Test and Evaluation) estimated that 7 out of 10 major weapon systems currently in development are encountering software problems, and the rate is increasing.[Ref. 11]

While the actual effects of software problems vary among the MCCR programs, most result in significant impacts on cost, schedule, and performance. A study conducted by the SEI estimates that the average MCCR program is likely to deliver one and one half years later than estimated at contract award. [Ref. 9] Past records of MCCR programs show that it is not uncommon for software development costs to exceed the original estimate by anywhere from 50-100%. [Ref. 6] Though performance problems cannot be as easily summarized, numerous reports by the GAO over the past several years have clearly detailed the impact of performance shortfalls on a program's life-cycle and show that they contribute directly to cost and schedule problems. Whether the results impact cost, schedule, or performance, they always bring increased visibility to a program. This negative visibility often does the most damage or provides the biggest threat to a program.

4. Why Does This Happen?

Over the past two decades numerous reports and studies have analyzed software development in the DOD in an effort to identify the software related problems and address their causes. GAO reports have done much of the work in identifying the software problems that plague so many major weapon programs. In December, 1992, a GAO report titled *Mission Critical Systems, Defense Attempting to Address Major Software Challenges* provided an overview of these reports that summarized the common software development problems and reported what DOD was doing to address them.[Ref. 12] DOD has also conducted numerous studies that have attempted to address the causes of these problems. These studies have been much broader in scope, focusing on issues of problems associated with acquisition policy, the software acquisition process, management oversight, and the supply of qualified personnel, to name a few. [Ref. 14]

Yet, even with such detailed visibility and analysis on software problems and their causes, the majority of these problems still exist. Correcting them has proven to be difficult. This difficulty persists for several reasons.

First, the software required to operate mission critical systems is, by nature, extremely complex. This complexity comes from the missions these systems perform and the environment they must operate in. For example, performing functions such as running large, graphical command and control networks, defending against airborne attack, or integrating multiple sensor data requires state-of-the-art technologies. These systems must operate in a real-time environment (in itself an extremely difficult task), over large geographical areas, and often be able to interoperate with other complex systems. Additionally, they must continue to function during enemy attempts to destroy or disrupt them. Producing a system of this complexity and reliability is no simple task. [Ref. 14:p. 5]

Second, studies by the SEI indicate the majority of DOD agencies and defense contractors engaged in the acquisition, production, or maintenance of software are operating at an immature level of software development process maturity. Established by DOD to improve the practice of software engineering, the SEI developed a five-level process maturity model to assist and evaluate a contractor's software process. The SEI describes software process maturity in this way:

In a mature software process, people, methods, techniques, and technology are effectively and efficiently coupled to consistently produce quality software within the constraints on cost and schedule requirements. In an immature software process, costs and schedules are largely unpredictable, quality is generally marginal, and technology is often used ineffectively. [Ref. 13]

Pictured below, Table 2 depicts SEI's Software Process Maturity Model [Ref. 15: Fig.1.2.1], and Figure 3 displays the distribution of software process maturity across the contractors and programs assessed by the SEI as of 1991 [Ref. 14]. Note that the vast majority of contractors (81%) were assessed at the most immature level (level 1) and the highest level achieved at that time was level 3.

TABLE 2. SEI SOFTWARE PROCESS MATURITY MODEL

Level	Characteristic	Key Problem Areas	Result
5 **Optimizing**	Improvement fed back into process	Automation	Productivity & Quality
4 **Managed**	(quantitative) Measured process	Changing technology Problem analysis Problem prevention	
3 **Defined**	(qualitative) Process defined and institutionalized	Process measurement Process analysis Quantitative quality plans	Risk
2 **Repeatable**	(intuitive) Process dependent on individuals	Training Technical practices - reviews, testing Process focus - standards, process groups	
1 **Initial**	(ad hoc/chaotic)	Project management Project planning Configuration management Software quality assurance	

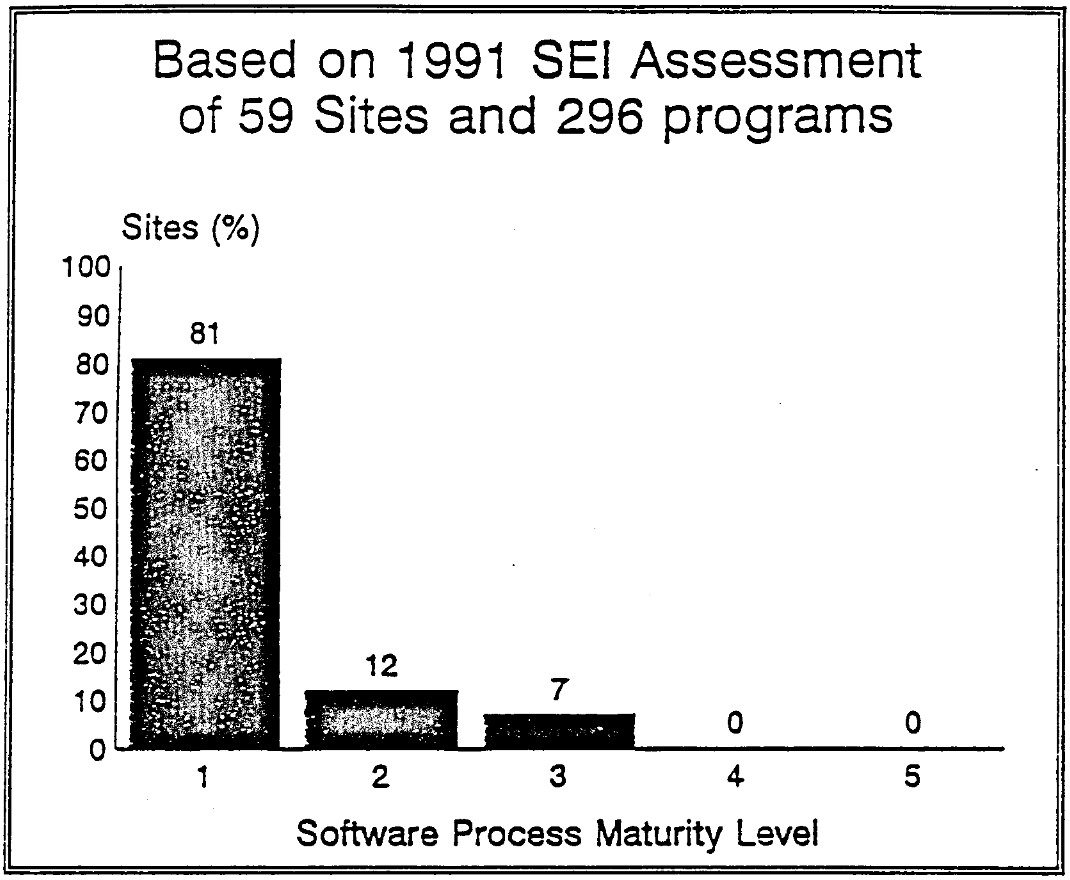

Figure 3. Software Process Maturity Distribution

Finally, the growing complexity and software dependence of current mission critical systems is outpacing the understanding of software as a product and its development process. As a result, weapon systems are continuing to be designed with more complicated software while developers are still wrestling with accurate methods of measuring critical software characteristics such as performance, security, and correctness, and the progress of developing software products. [Ref. 8] This situation is fueled by the DOD's runaway demand for new, leading edge weapon systems which grows far faster than the supply of skilled software professionals. [Ref. 10]

These factors combine to make the development of the unique and complex software required by weapon systems a high risk undertaking. The normal cost and schedule constraints of a DOD program weapon system software development, often unprecedented in design, presents an enormous challenge for software programmers and managers. The president of one software programming company supports this in his own words:

> Whenever I see one of these major new programs get behind schedule or go over budget, I know that the programmers just took the gamble and lost. It's embarrassing as a member of this industry to admit that these software projects are so risky that it's impossible to accurately predict a budget or schedule, but that's the case. [Ref. 10:p. 5]

Most studies agree that there are no easy answers to these shortfalls in software development. While the DOD has been making efforts to address the issues, the current state of software development is still marked by high risk. A weapon system today that requires development of a complex software system must do so in a development process that is fraught with problems.

D. CURRENT DOD PROCUREMENT ENVIRONMENT

The risk of complex software development is not the only challenge facing program offices. Contemporary political and socioeconomic forces often make the very environment they operate in a precarious one.

The end of the Cold War and strong domestic concern over the national deficit have prompted significant budget reductions for the DOD. Historically, Procurement and Research, Development, Test and Evaluation (RDT&E) accounts mirror the trend of the total DOD budget, as they are the most discretionary. [Ref. 15] As shown in Figure 4, this has resulted in a consistent decrease in the annual Procurement and RDT&E (6.3 & 6.4) accounts since 1985. [Ref. 18, p. 8]

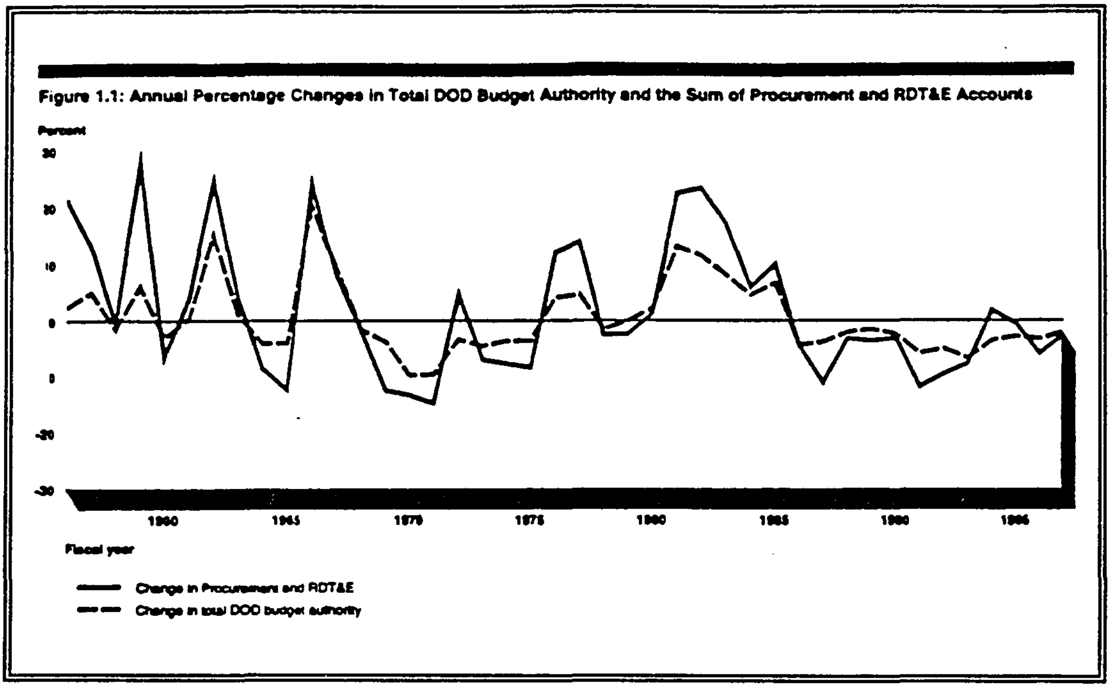

Figure 4. Annual Percent Change in Total DOD Budget Authority and the Sum of Procurement and RDT&E Accounts

Now program offices must plan, develop, and produce sophisticated weapon systems under a much tighter monetary constraint. Affordability becomes a major issue facing all weapons programs. The affordability of a weapon system is directly affected by cost, schedule, and performance problems confronting these systems. If systems become unaffordable, politically or economically, they risk termination.

As the constraints on program offices have increased over the years, so has Congressional oversight of the procurement system. DOD procurement decisions commit the nation to billions of dollars in current and long-range expenditures. This has a large impact on the resources available to other national priorities. Because of the size and importance of these commitments and a history of weaknesses in the DOD acquisition system Congress closely scrutinizes the procurement process. The GAO, primarily responsible for this auditing action, has completed over 900 reports and testimonies since 1971 covering every aspect of weapon system acquisition. [Ref. 18:p. 11] GAO reports that surface problems or inefficiencies in a weapon system procurement program bring high visibility to the program office and may cause loss of congressional support for a program.

The procurement environment of decreasing funds and increasing scrutiny has different effects on each program office. Some programs may be partially shielded from this environment based on the priority of the mission need they satisfy. But, regardless of priority, all programs must still function within this environment where cost, schedule, and performance problems can threaten future funding decisions.

E. CORRECTIVE SOFTWARE MANAGEMENT

In relation to the procurement of MCCR systems, corrective software management encompasses the decisions and management actions necessary to rectify problems encountered in the development of weapon system software. The primary goal of corrective software management is to satisfy system requirements while minimizing impact on cost and schedule. When executed effectively, it entails a planned event focused specifically on the correction of software deficiencies with the necessary resources committed to accomplish and verify the action.

Although one may assume this would be a common management response to a problem, GAO reports from the past two decades describe numerous weapon system programs experiencing years of delayed fieldings and substantial cost overruns when software deficiencies were not effectively addressed. For example, in a March 1993 GAO testimony before Congress the Director of the Defense and Security Information Systems reported on the status of the C-17 Aircraft's embedded software. The program encountered significant software problems early in development and was unable to deliver the proper software for the initial test flight. Rather than correcting these deficiencies before proceeding, the contractor was allowed to defer software development to follow-on test flights and use developmental shortcuts in an effort to stay within the original schedule. This strategy continued to delay full testing of the software, forced changes in the acquisition plan, and precluded demonstration of the aircraft's ability to meet requirements. The software has remained behind schedule in the program, cost overruns have been significant, and performance shortfalls have hampered the system. [Ref. 16]

Corrective software management, when used effectively, should focus on bringing the software of a system up to the developmental standard planned for when the major deficiencies were identified. During this effort a secondary goal should be to minimize deviation from the development schedule and adhere as closely to the original acquisition plan as possible. This requires accepting and committing the additional funds and time necessary to accomplish the task of correcting the software problems at hand before any further work requiring that software is attempted. There little historical evidence that shows this approach has been used often in DOD procurements, but it has happened at least once. That one specific case of corrective software management is the focus of this thesis.

F. SUMMARY

This chapter has provided the background for the complex and challenging environment in which DOD programs procure software. Software plays a vital role in virtually all major defense weapon systems today. The annual cost and volume of software demanded by the DOD has grown exponentially in the last three decades, but this growth has been accompanied by growing software development problems. these problems have had a damaging effect on the cost, schedule, and performance of many weapon system programs and is now a major concern in the DOD. The DOD's ability to effectively manage the development of this software has been outpaced by the rapid growth in the size and complexity of the software in today's sophisticated weapons. In the current, unforgiving procurement environment a program office should be prepared for the common occurrence of software development problems by understanding how to effectively use corrective software management.

Next, chapter III will discuss the case study methodology used in preparing this thesis.

III. METHODOLOGY

A. GENERAL

The primary research strategy used in creating this work was the case study. This chapter addresses the case study methodology. A clear description of the case study as a strategy and a discussion of its advantages and drawbacks will be provided. The chapter will also discuss the application of the case study strategy to this thesis.

B. CASE STUDY AS A RESEARCH STRATEGY

When used as a research strategy, the case study can be technically defined as follows:

A case study is an empirical inquiry that:

- investigates a contemporary phenomenon within its real-life context; when
- the boundaries between phenomenon and context are not clearly evident; and in which
- multiple sources of evidence are used [Ref. 17:p. 23].

This definition highlights the characteristics that make the case study so useful in researching social science issues. It lends itself as an appropriate research strategy to be used in many settings in this area. They include:

- policy, political science, and public administration research;
- community psychology and sociology;

- organizational and management studies;
- city and regional planning research, such as studies of plans, neighborhoods, or public agencies, and
- the conduct of a large proportion of dissertations and theses in the social sciences [Ref. 17:p. 13].

Of course, there are other methods available to conduct this type of research. The case study is one of five common social science research strategies. The remaining four are experiments, surveys, histories, and the analysis of archival information (as in economic studies). Each of these strategies has a distinguishing set of characteristics that make it more suitable than the others for researching certain situations. The suitability of a strategy to a given situation depends on three conditions: (1) the type of research question being posed (who, what, where, how, or why), (2) the extent of control an investigator has over actual behavioral events. and (3) the degree of focus on contemporary as opposed to historical events. Table 3, below, displays how each of the five major strategies relates to relevant situations based on the three conditions. [Ref. 17:p. 17]

Table 3 shows that, while there is some overlap of applicable strategies for a given condition, each specific combination of the three conditions together is best suited to a specific research strategy. For example, experiments, histories, and case studies are all strategies that apply to "how" or "why" forms of research questions in the first condition. Yet, while an experiment focuses on contemporary events, it requires control over the behavior of those events. Histories do not require control over behavioral events, but focus on events from the past. This leaves the case study with its own niche as the preferred strategy when "how" or "why" research questions are asked, when the investigator has little control over events, and when the focus is on a contemporary event within some real-life situation. [Ref. 17:p. 19]

TABLE 3. APPLICATION OF DIFFERENT RESEARCH STRATEGIES

Strategy	Form of Research Question	Requires Control Over Behavioral Event?	Focuses on Contemporary Events?
Experiment	how, why	yes	yes
Survey	who, what, where, how many, how much	no	yes
Archival Analysis	who, what, where, how many, how much	no	yes/no
History	how, why	no	no
Case Study	how, why	no	yes

Enumerating the advantages of the case study strategy shows its value to appropriate research requirements. As a research technique, "...the case study contributes uniquely to our knowledge of individual, organizational, social, and political [events]" [Ref. 17:p. 14]. As mentioned earlier in this section, this implies a wide range of research situations to which the case study strategy is ideally suited. This specific need for case studies arises from the desire to understand complex social events. Using the strategy enables an investigation to capture the holistic and meaningful characteristics of these events and analyze them as they occurred in their environment. [Ref. 17:p. 14]

Another advantage of the case study is its access to a wide variety of evidence [Ref. 17:p. 20] Because the focus of case study research is on contemporary issues, the sources of data are normally plentiful. These sources include current and historical documents, data files, artifacts, observations, and interviews. Unlike histories or surveys, the ability to conduct personal interviews and make personal observations is a particular strength of the case study. This variety of data sources is important to conducting an accurate, unbiased case study. Rarely does one source of data contain all the pertinent evidence required to clearly understand a complex situation. In a case study the investigation can gain insight from several perspectives and better piece together the whole picture.

Related to this is another advantage that stems from the opportunity to conduct personal interviews with people directly involved with the event being studied. The attitudes, personal expressions and comments of key players absorbed during interviews cannot normally be gleaned from reading documentation alone. The richness of this type of data provides a unique insight into the real meaning of events and relationships within their own settings. [Ref. 18:p. 8]

The case study strategy does have potential drawbacks. Some of these drawbacks stem from the same characteristics that provide advantages. For instance, while personal interviews can add rich, qualitative data to the research, they can also add biased views that can influence the investigator. Similarly, lack of data sources may yield a one-sided view of a situation that strongly influences the findings and conclusions. The investigator must follow a rigorous plan while conducting the research in order to ensure alternate views or opinions have been heard.

Another drawback and common complaint about case studies is that they provide little basis for scientific generalization. This is primarily the case when only a single case study is performed. This complaint centers on the belief that single case studies cannot provide enough substantial evidence to validate the hypothesis of the research. This is a valid argument when case study results are statistically generalized to a population. There is little confidence in the statistical findings of any single case. Investigators can avoid this drawback by using the case study to generalize theoretical propositions. Case study findings can be used to generalize theory with reasonable confidence. [Ref. 17:p. 21]

Often in research there is a preference for being able to control events which brings up the subject of replicability. Much of the data compiled in case studies are qualitative, gathered through interviews and observations. Analysis of qualitative data can lead different investigators to different conclusions. The conduct of a specific case study is difficult, or often impossible, to replicate. This situation adds a degree of uncertainty to the case study research process. [Ref. 18:p. 16]

These drawbacks do not necessarily detract from the value of using the case study as a research strategy. Each research strategy has its own advantages and disadvantages depending upon the conditions they are used in. The investigator must take care in the design and conduct of a case study to avoid these drawbacks and follow standardized case study techniques. If used properly, the case study strategy can yield reliable, valid results.

C. APPLICATION OF THE CASE STUDY STRATEGY TO THIS THESIS

This particular research topic was well suited for the case study strategy. The research question is a "how" question. The primary research question developed for this study is: "How can a Program Manager most effectively use corrective software management to solve problems that develop in the software acquisition process?" Because the issue involved the actions of an Army program office, the investigator had no control over behavioral events. Obviously, in referring to software management in an existing Army program office, the focus of the research is on contemporary events. Thus, the three conditions for case study suitability were clearly satisfied.

The case that is the subject of this study involves a three year period in the life of an Army project office developing a battlefield command and control system. The case describes the policies, decision process, and actions in the correction of serious system software deficiencies discovered during the engineering and manufacturing development (EMD) phase. The events pertinent to the case research include the initial discovery of software deficiencies, the analysis of the problem, the planning of corrective action, the conduct of corrective action, and the verification of the corrections. The corrective action plan was successful and will be discussed in the next chapter.

The contemporary issue involved is the effectiveness of program offices in managing the correction of software deficiencies that hinder the development of the system. The significance of the problem has grown in the last two decades. It is an issue that affects many DOD program offices today.

The project office in this case still exists and agreed to assist the research effort. Several sources of data were available and included written documentation, personal interviews, and taped interviews. The written documents included Government reports on the system, Government decision memorandums, and Government and contractor briefing charts. Interviews were conducted with Government project office personnel, Government contracted support personnel, and prime contractor personnel. These included the project managers from the Government and prime contractor, technical advisors and engineers from the Government (contracted support) and the prime contractor, and the Government Program Executive Officer having oversight of and decision authority over the project. The investigator was allowed access to both the Government project office and the prime contractor's program office.

D. SUMMARY

This chapter has discussed the use of the case study as a research strategy. It has highlighted some strengths and weaknesses of the strategy and described the suitability of the strategy's application in this thesis. The use of this methodology is advantageous here because of the significant results of this one project office in the area of corrective software management. The goal of the case study strategy in this work will be to illuminate the decisions that were made, why they were made, how they were implemented, and what results occurred. In the next chapter the case will be described in detail.

IV. THE ENHANCED POSITION LOCATION REPORTING SYSTEM

A. GENERAL

To win on today's battlefield you must first win the information war [Ref. 19]. This concept has fostered a need for a complex network of battlefield data distribution and communication systems in the Army. Many of the software intensive systems required to meet this need are currently in development by Army program offices. The Enhanced Position Location Reporting System (EPLRS) is one of these systems. This chapter will describe the EPLRS system and give a brief look at its developmental history. It will also describe the EPLRS Government and contractor program offices. The chapter will conclude with a review of the program schedule through completion of technical testing in 1989 and detail the significant problems encountered during this testing.

B. THE EPLRS SYSTEM

EPLRS is a computer based communications system which provides secure, jam resistant, near real-time data communications to high priority data subscribers. Additionally, it provides identification, navigation, position location, and automatic reporting for tactical forces. Its primary mission is to provide this data communication capability in support of the data subscribers of the Army Tactical Command and Control System (ATCCS). [Ref. 20:p. 1] This requires EPLRS to interface with and distribute and relay data for all of the automated battlefield systems that make up the ATCCS. ATTCS includes systems such as Advanced Field Artillery Tactical Data System (AFATDS), Forward Area Air Defense Command, Control and Intelligence (FAADC2I), All Source Analysis System (ASAS), and Maneuver Control System (MCS). Additionally, it must maintain its own network operation for its position, navigation, and identification role. This volume of system interfaces, near real-time function performance and network management requires EPLRS to have extremely complex software. Altogether, the system requires over 500,000 source lines of code (SLOC) to perform its mission [Ref. 21].

1. Description

EPLRS consists of a network of ultra-high frequency (UHF) radio systems managed by a net control station (NCS). The radio system can be carried by individual soldiers, mounted on combat vehicles, or installed in aircraft. The NCS, housed in truck mounted shelters, provides network management functions for the entire EPLRS system. Figure 5 depicts the basic EPLRS elements [Ref. 22:p. 1].

EPLRS is a spread spectrum system which utilizes Time Division Multiple Access (TDMA) technology. It is protected against electronic counter measures (ECM) through the use of frequency hopping and adaptive relay techniques, and security transmit power anti-jam characteristics. The system provides two levels of secure capability using cryptographic devices located within each radio set. The NCS is capable of storing and distributing cryptographic keys for each net in operation using over the air rekey

(OTAR). The EPLRS radio sets are designed to have a simplex data throughput rate of up to 1200 bits per second (bps). [Ref. 31:p. 2]

EPLRS position location function is designed to locate an airborne user station within 25 meters of its actual location and a ground based user station within 15 meters of its actual location. The system is interoperable with the Marine Corps Position Location Reporting System (PLRS). EPLRS can support both Army Data Distribution System Interface and MIL-STD-1553B interface protocols. These two standard host system data interfaces allow EPLRS to interface with critical existing and emerging battlefield data systems. [Ref. 32:p. 2]

2. Development History

During the 1970's the Army and Marine Corps were developing PLRS as a joint program. Hughes Aircraft Company (hereafter referred to as Hughes) was the prime contractor for this system. PLRS was being designed to provide field commanders information about the location of their forces and other friendly units in their area of operation. While PLRS was being developed, a need for a method of exchanging data among a variety of automated battlefield data systems evolved in the Army. In 1978, Hughes was awarded an Army contract to study the feasibility of integrating PLRS with the Joint Tactical Information Distribution System (JTIDS) to satisfy this need [Ref. 20:p. 9]. The Army chose to continue with this concept and initiated a program in July 1979 known as the PLRS/JTIDS Hybrid, commonly called PJH.

In the mid 1980s the Army restructured the PJH system architecture and separated JTIDS from the program. The system was renamed the Enhanced Position

Location Reporting System (EPLRS). EPLRS and JTIDS then became separate products under a project called the Army Data Distribution System (ADDS). The ADDS Project Manager (PM ADDS) currently manages the acquisition of PLRS, EPLRS, and the Army portion of JTIDS. [Ref. 22:p. 1]

The resulting system, EPLRS, now provided two primary functions. One was the basic PLRS function: providing position location, navigation, and identification information and reporting that information to battlefield commanders upon request. The other function, representing the enhancement, provided automated network data communications support to weapon systems, sensors, and command, control, communication, and intelligence (C3I) elements. This function equates to providing a data pipeline for several battlefield host data and communication systems. [Ref. 31:p. 2] This system was developed as an acquisition category (ACAT) 1D program.

C. THE EPLRS PROGRAM OFFICES

This section will detail the program office structure of both the Government and contractor sides of the EPLRS program. While these offices may have changed since the inception of the PJH program, this discussion will pertain to the program office structure since 1988.

1. The EPLRS Government Program Office

The EPLRS program office is located in Fort Monmouth, New Jersey. Because EPLRS is a product of the ADDS program, PM EPLRS falls under the control of PM ADDS. PM ADDS is managed by the Program Executive Office, Communications Systems (PEO COMM). Associated with the EPLRS and ADDS program offices is the office of the TRADOC Systems Manager, Army Data Distribution System (TSM ADDS) in Fort Gordon, Georgia. TSM ADDS is the liaison between the user community and the program office.

The EPLRS program office has a core staff of three people: the Product Manager (Lieutenant Colonel/0-5), the Deputy Product Manager (GM-14), and a secretary. The remainder of the staff in the EPLRS office comes from the matrix support structure of the Communications and Electronics Command (CECOM). While the total number of people on the EPLRS staff has varied, the average strength is 16 personnel. The resident matrix personnel for EPLRS are a combination of Government employees and Government contracted employees. The mix of these employees varies with the availability of personnel in the matrix support organization. On average there has been a two person staff responsible for the program's software. Their responsibility is to oversee the development and production of all software associated with the system and ensure compliance with the applicable software plans and specifications.

In addition to the main office in New Jersey, the EPLRS organization has a field office located at the prime contractor's facility in California. Known as the California Field Office (CFO), it also has a core staff of three personnel: the CFO Chief (Lieutenant Colonel/0-5), Deputy Chief (GS-13), and a secretary. The remainder of the CFO staff is a combination of government and contracted engineers and technicians. The total number of CFO personnel has varied during the program, but has roughly averaged 15 personnel. The responsibility of the CFO is to provide on-sight management of the program for the PM and monitor the work of the contractor. As an on-sight representative of the PM, the personnel in the field office can provide immediate feedback to the contractor and relay information between the contractor and the PM.

2. The EPLRS Prime Contractor Program Office

The prime contractor for the EPLRS program is Hughes Aircraft Company. The actual system development work is performed by the Surface Systems Division of the Aerospace and Defense Sector of Hughes located in Fullerton, California. During the time of this case, the Hughes EPLRS and PLRS program management offices (PMO's) were under the control of the Tactical Data Distribution Systems PMO. The EPLRS PMO was supported by a matrix support system at Hughes similar to the Government matrix support system. Like its Government counterpart, the EPLRS PMO at Hughes relied on software technicians from the Hughes software engineering directorate (SED) for the programs software work.

D. EPLRS ACQUISITION STRATEGY

The plan to procure EPLRS, initiated as PJH, was conducted under a five phase development strategy. This tailored strategy reflected the fact that the program was incorporating existing technology from other programs still under development. The five phase development strategy was as follows:

1. A feasibility study of the concept of integrating PLRS and JTIDS, contracted to Hughes, was successfully completed in mid 1980. This work was identified as "System Definition" and became Phase 1 of the development strategy.

2. In June 1980, Hughes was awarded a contract to demonstrate the technical feasibility of the system and validate the concept. This effort became Phase 2 and was identified as "Interoperability Verification". Hughes completed the contract in mid 1982.

3. Phases 3 and 4 were consolidated and identified as "Operational Prototype". The contract for Phase 3/4 was awarded to Hughes in March 1982. The purpose of this phase was to develop an operational prototype, upgrade the system software, and develop the interface with other systems. Early in this phase (September 1982) the Army System Acquisition Review Council (ASARC) approved acceleration of the five phase development strategy for PJH. Towards the end of Phase 3/4 the PJH architecture was restructured and JTIDS was separated from the program. The system was renamed the Enhanced Position Location Reporting System (EPLRS). Phase 3/4 was completed in February 1987.

4. The Phase 5 contract was awarded to Hughes in April 1985. This contract required Hughes to produce sufficient quantities of equipment for developmental and operational testing. This would be accomplished by modifying PLRS radio sets and master stations to the required PJH, and later EPLRS, configuration. Phase 5 would culminate in the production of four EPLRS NCS's, 211 Enhanced PLRS User Units (EPUU's), and six electronic test sets for use in Prototype Qualification Test-Contractor (PQT-C) and Government technical testing. Thus equipment was to be delivered in a series of deliveries scheduled from November 1987-May 1988. [Ref.31]

Based on early success in the accelerated program strategy, a Department of the Army In-Process Review held in January 1987 approved a Low Rate Initial

Production (LRIP) of an additional eight NCS-Es and 1843 EPUU's. These systems would be used to support future testing and initial fielding requirements. Contingent on success in Initial Operational Test and Evaluation (IOT&E) full scale production was scheduled to begin in late 1989 [Ref. 20]. An early acquisition plan, shown in Figure 5, depicts this strategy [Ref. 23].

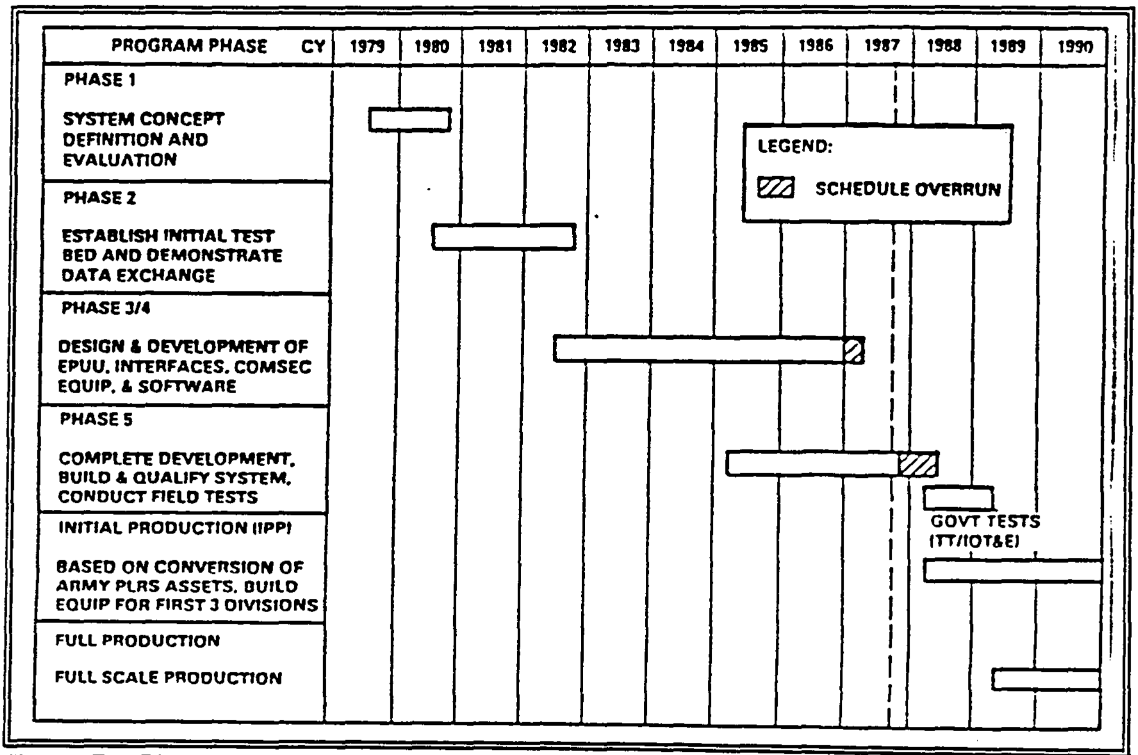

Figure 5 EPLRS Program Plan, Nov. 1987

E. TECHNICAL TEST II

Hughes conducted formal test and demonstration of early prototype units in 1986. After making desired changes, formal prototype qualification testing (PQT-C) was conducted by Hughes on pre-production equipment in 1987. This PQT-C, as the term was used by Hughes, consisted of tests and demonstrations which confirmed the integrity of the system design over the specified operational and environmental range. This series of tests included software verification, performance verification, functional certification, interface verification, environmental tests, electronic warfare tests, electromagnetic interface and compatibility tests, reliability qualification tests, and a maintainability demonstration. [Ref. 24]

This was followed by the Government's version of prototype qualification testing (PQT-G), scheduled to begin in early 1988. PQT-G, a term also used by Hughes, refers to technical testing conducted at Government test facilities under operational conditions. These tests verify the performance specifications, objectives, producibility, adequacy of technical data package, and supportability. It also considers safety, health hazards, human factors, and manpower and personnel integration (MANPRINT) aspects. This Government testing is required to support IOT&E readiness and production decisions. For the EPLRS program this was referred to as Technical Test II (TT-II). TT-II was executed by the U.S. Army Test and Evaluation Command (TECOM) at the U.S. Army Electronic Proving Ground (EPG), Fort Huachuca, Arizona. The testing resulted in two phases running from May 1988 to April 1989. The remainder of this section will explain the goals and results of TT-II.

1. Test Goals and Structure

For the EPLRS program, TT-II was an important, closing phase of developmental testing. The objectives for TT-II were to determine whether EPLRS could meet its technical requirements in the intended operational environments and whether it was ready to proceed into operational testing. Additionally, TT-II would serve as input in determining whether the system should be granted material release and proceed into full scale production at a later date.

As originally planned, TT-II would be conducted across a five month period. Another five months at the test site were allocated for follow-on design modifications, retests, and administrative actions. Due to the complexity of the EPLRS system and the numerous technical requirements to be assessed, the technical testing would be conducted across several, separately run scenarios. Each scenario would assess a different group of requirements. The detailed test plan would simulate varying communication traffic densities across a network of operational stations. This would include the EPLRS requirement to interface with several other "host" communication and data systems. The interface requirement posed one of the largest challenges of TT-II for both the EPLRS Program Office and the testing community at the EPG. While a critical requirement for EPLRS is to successfully interface with other battlefield information systems (AFATDS, FAADC2I, ASAS, MCS, and others), most of these systems were still under development and not fielded prior to TT-II. The test agency at the EPG, was responsible for producing the host simulators that could replicate the real systems during interface tests. Following successful completion of TT-II, the EPLRS equipment would be shipped directly to Fort Hood, Texas to begin operator training and, subsequently, IOT&E.

2. TT-II Deficiencies

TT-II identified numerous, significant deficiencies with the EPLRS system and test plan. The initial phase of TT-II was conducted from May '88 - September '88. At the beginning of the test it was recognized that the test instrumentation software needed to simulate the various battlefield information system interfaces was not fully developed. The decision was made to use the instrumentation in its current condition. This added a larger than normal component of error to the test data. The resulting statistical estimates of system parameters could not provide accurate estimates of system performance for test analysis.

The EPLRS equipment itself revealed significant deficiencies in both capability and reliability. To summarize, the problems areas included failure of the system to efficiently build a network of users, NCS and EPUU reliability, the ability to establish communications with stations controlled by other NCS's (intercommunity needlines)[1], and the ability to rebuild a network after disruptions. The majority of these problems were a result of software and firmware deficiencies. This is substantiated by the sole concluding statement of the test directorate's written results of TT-II, Phase I:

---

[1] For the EPLRS system a *needline* is "A requirement for two or more users to communicate. Needlines are defined by a source, destination, rate, [and] priority...". A community refers collectively to the EPUU's, or users, serviced by one NCS.

"It was concluded by the PM that the system should undergo firmware and software revisions, and a second phase of its Technical Test should be conducted in early 1989" [Ref. 25]. The EPLRS PM proceeded with plans to conduct TT-II, Phase II.

In the four months that followed test instrumentation was improved and software and firmware revisions were made to the system. EPLRS proceeded into TT-II, Phase II at EPG in January 1989 and successfully demonstrated a number of requirements. Nonetheless, the test continued to surface major problems showing that significant deficiencies still existed in the system. Although the test instrumentation performed satisfactorily, critical problems continued to center around the developmental model of the NCS and the maturity of the system's software. The primary Government concerns resulting from phase II of TT-II were: intercommunity needline performance, system reliability, system and software maturity, human interface issues, and general system performance [Ref. 26].

Once again, the primary source of these system deficiencies was software and firmware problems. As a result, the EPLRS system did not successfully pass TT-II.

3. Impact of TT-II

The results of TT-II caused the program to breach several thresholds for performance criteria. This deviation from the EPLRS program baseline required a Deviation Report be submitted by the PM. Additionally, a Program Deviation Report Review Panel, headed by the Deputy Under Secretary of the Army (Operations Research) (DUSA(OR)), was formed to review the EPLRS program. The increased visibility placed the program at risk of cancellation. In an effort to ensure survivability of the program, the EPLRS PM was forced to restructure the program with a corrective action plan that would satisfy the review panel's concerns over whether the deficiencies could be fixed. The resulting plan to restructure the EPLRS program was accepted by the review panel. The panel recommended to the Army Acquisition Executive that the new EPLRS program baseline be approved. [Ref. 27] The course of events brought major changes to the EPLRS acquisition plan.

F. SUMMARY

This chapter has introduced the EPLRS program and the agencies responsible for it. It also provided a brief look at the history of the program and the acquisition strategy. The original EPLRS program strategy had a high level of concurrency. This level of concurrency and the decision to accelerate the developmental strategy increased the program risk. As a result of failing TT-II, the program was restructured to resolve critical deficiencies.

The next chapter will review the restructured program in depth. The corrective action plan and its implementation by the Government and contractor program offices will be covered in detail. The role and impact of program oversight agencies and other external agencies will also be discussed.

V. EPLRS CORRECTIVE ACTION

A. GENERAL

As described in the last chapter, the results of TT-II Phase II presented a serious challenge to both the Government and contractor EPLRS program offices. They responded with indepth analysis and planning that produced an aggressive corrective action plan. This chapter will focus on the corrective action used by the Government and Hughes to correct system deficiencies identified in technical testing.

First, the chapter will review the Government guidance in developing the plan. Next, it will provide a detailed review of the corrective action as it was executed. Lastly, the chapter will discuss the key actions taken by both the Government and Hughes Aircraft Company that were instrumental in the success of the corrective action.

B. GOVERNMENT GUIDANCE

Three important actions by various Government players served to shape the eventual corrective action plan for the EPLRS program. These were the program restructuring plan proposed by PM ADDS, the new test criteria developed by the TSM, and the report of the Program Deviation Report Review Panel. These actions set the boundaries and criteria for the corrective action plan.

1. Program Restructuring Plan

EPLRS deficiencies in TT-II revealed immature software, firmware, and hardware that could not adequately meet user requirements. This problem was primarily the result of an accelerated and concurrent program schedule. The intent of the restructuring plan was to adopt a more conservative approach, satisfy testing requirements, and return the program to a non-concurrent schedule. The restructuring plan was designed to accomplish the following [Ref. 28]:

> 1. *Fix and demonstrate technical test corrective actions.* This entailed the actual execution of a corrective action plan. It required production representative models for testing, a process to verify correction of TT-II faults, and Government participation and witnessing.
>
> 2. *Reduce technical risk through block upgrades.* This would be accomplished through an evolutionary strategy. Low risk enhancements to the system would first be used to put a qualified system in the field. Higher risk enhancements would be added as block upgrades to put a better system in the field.
>
> 3. *Protect the Government's cost exposure.* Initially, the Government would only guarantee payment for labor to correct system deficiencies. Hardware costs would only be paid for after successful completions of performance demonstrations. Clearly defined progress points would be established to monitor progress and base progress payments from.
>
> 4. *Test what [the Government] intends to buy.* Conduct corrective action and future testing on production representative units rather than engineering development models.
>
> 5. *Consider user schedules.* In restructuring, try to minimize the impact delayed fielding will have on units scheduled to receive EPLRS.

2. TSM Development of New Test Criteria

Based on EPLRS' performance in TT-II, the TSM and the user community were losing confidence in the contractor. The TSM was concerned with Hughes' ability to solve the serious technical problems in a reasonable time period. After an analysis of the test results and the work estimated to fix the system, the TSM provided PM ADDS with a set of future test thresholds which would raise the user's confidence in system technical performance. Specifically, these thresholds were test criteria identified for the major areas of system deficiencies which the TSM wanted to see met in upcoming tests. These criteria were set to account for incremental improvements beginning with corrective action testing and continuing through follow-on technical testing and IOT&E. [Ref. 26]

3. Program Deviation Report Review Panel

This panel, chaired by the DUSA (OR), was required by Title 10 of the U.S. Code to review the EPLRS program after it breached both cost and schedule areas of its program baseline. The panel reviewed several inputs: a technical evaluation of the EPLRS software system and the capability of Hughes's software/firmware development process, technical and program presentations from Hughes employees, the PM's proposed restructuring plan and new program baseline, and the Independent Operational Assessment of EPLRS conducted during TT-II. The panel's conclusions and recommendations, as forwarded to the Army Acquisition Executive, would form the guidance the EPLRS program must adhere to for its survival. The conclusions of the panel are summarized below [Ref. 27]:

1. There are two flaws in EPLRS performance which, if not corrected, will be fatal: 1) the failure of the system to establish intercommunity needlines, and 2) the demonstrated reliability of the NCS.

2. The software/firmware efforts of the contractor should be focused on demonstrating that successful operation in a multiple NCS configuration is possible at full EPUU load.

3. The corrective action phase of the program's efforts should be clearly separated from the modernization phase currently in progress. The panel stated that the first step in "getting well" must be directed towards achieving a successful system demonstration unencumbered by a simultaneous attempt at system modernization.

4. The Project Manager's restructured program plan was adequate to ensure Government visibility into contractor progress and control Government financial exposure.

5. A maturity matrix must be developed and included in the Test and Evaluation Master Plan (TEMP) and program baseline. This would be instrumental in approving future program events based on successful resolution of current problem areas in the system.

6. Development of an adequate simulation program for development and testing purposes be undertaken as an urgent matter.

These three actions clearly outlined the challenge facing the EPLRS program in correcting the problems identified in TT-II. They would provide the guidance for development and award of the contractual agreement used to execute corrective action for the program.

C. PRODUCTION SYSTEM VERIFICATION

The actual corrective action plan developed between the Government and Hughes was called Production System Verification (PSV). This section will provide a description of the concept, the basic plan and schedule, and how successful demonstrations in the plan triggered follow-on decisions.

1. Concept

The concept for PSV was initiated by the PM ADDS and his Hughes counterpart, the TDDS PMO, with the original intent of showing that correction of TT-II deficiencies was feasible. The concept called for a discrete, dedicated plan focused exclusively on correcting test deficiencies. The plan would center on a disciplined Test-Analyze-And-Fix (TAAF) methodology that emphasized risk management. The plan received top corporate involvement and high priority for resources from Hughes. The Army awarded PSV as a separate fixed price contractual effort at a price of $8.75 million. The actual contract was a modification to the current LRIP contract for the EPLRS program. Hughes organized the effort as a separate program with its own program management office. The goals of PSV were to:

1. Replicate and characterize the problems from technical testing.
2. Understand their root causes and the fixes required.
3. Develop and execute a quantitative plan to correct all deficiencies.
4. Achieve the original intent of TT-II.

2. The Plan

The PSV plan was based on a Test-Analyze-And-Fix (TAAF) methodology. As shown in Figure 6, the TAAF methodology called for an iterative field testing cycle which would characterize system problems and verify system performance after corrections [Ref. 29]. EPLRS would be operated in a scenario similar to that used in TT-II. The data collection process of each test provided the data required to conduct root cause analyses and develop corrective actions between field tests. There was an emphasis on strong configuration management in order to accurately track the numerous revisions to software and firmware associated with each test.

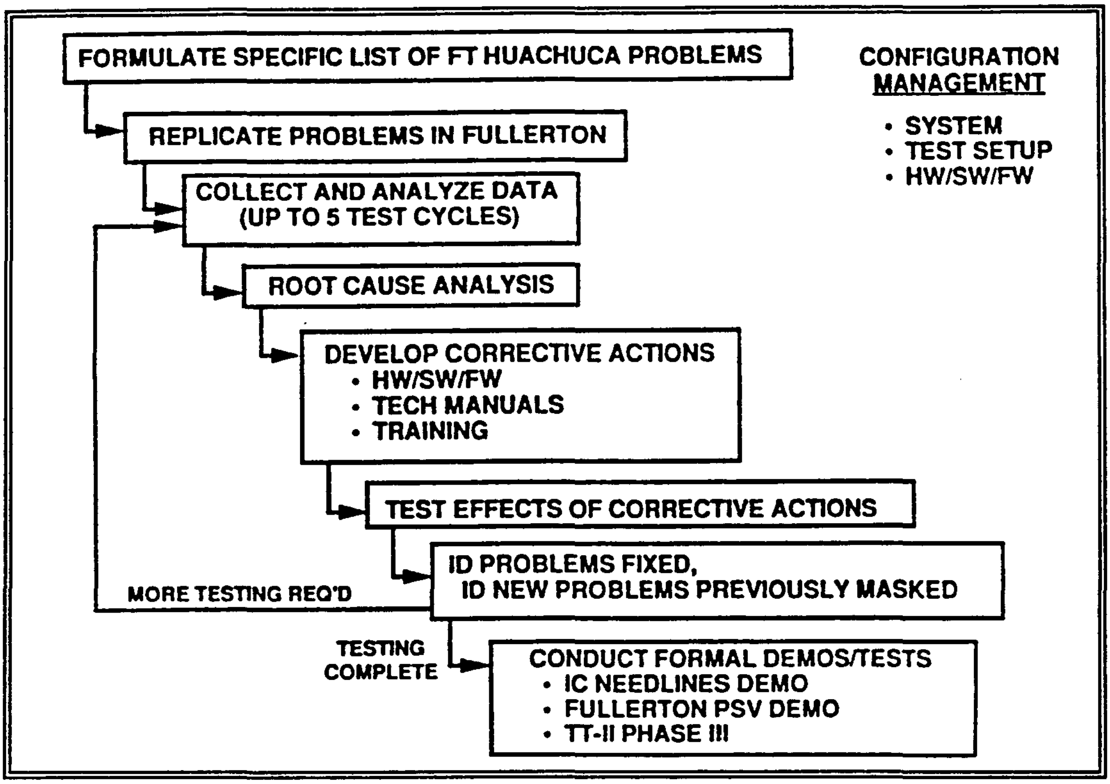

Figure 6 Test Analyze and Fix Methodology for EPLRS

The original plan included five field tests to be conducted in the area around Hughes' Fullerton, California plant. These "back yard tests" were contractor controlled and involved the deployment of EPLRS equipment at stationary and mobile sites throughout the community. The deployment was designed to simulate the typical operational dispersion of two combat brigades and the associated EPLRS unit density within that area. The U.S. Army Training and Doctrine Command (TRADOC) verified the accuracy of this deployment scenario. The test environment was designed to simulate operational field conditions and the actual EPG test environment as closely as possible.

While the PSV field tests were run by the contractor, the Government had several roles in the plan. The EPLRS PM and California Field office personnel monitored many PSV actions, including test preparation, testing, and analysis of results. EPG personnel were involved in test data collection and reduction. As mentioned above, TRADOC assisted with deployment verification and provided user participation for certain parts of PSV testing involving manpower and personnel integration (MANPRINT) deficiencies. Both Government and contractor personnel participated in IPRs conducted between deployments to assess status.

The general strategy for the PSV field tests (FTs) was first to replicate and characterize the problems found during technical test, incorporate the corrections to the system, and, finally, verify the proper system performance through demonstrations. This strategy, used with a disciplined engineering process and software and firmware configuration management, was geared to achieve an incremental increase in system performance. Figure 7, below, lists an overview of the objectives for each PSV field test. The objectives indicate the role of each field test in the strategy and show the software and firmware focus of the PSV plan [Ref. 29].

Based on the desire to minimize slippage of the EPLRS program, the PSV plan was assigned a compressed schedule. The PSV program actually began with FT #1 in June 1989, and would end with a formal system demonstration, scheduled for April 1990. The field tests, averaging 2-3 days in duration, occurred approximately every six

FIELD TEST #1

- SUCCESSFUL DEPLOYMENT
- BASELINING PERFORMANCE IN SOUTHERN CALIFORNIA EMI ENVIRONMENT
- REPLICATION OF TT PROBLEMS
- EVALUATION OF 15 CHANGES IN NCS SOFTWARE (FT. HUACHUCA TAPE VS FT 1 TAPE)

FIELD TEST #2

- MAJORITY OF KNOWN SW & FW FIXES INCORPORATED

FIELD TEST #3

- ARMY NCS OPERATORS UTILIZED
- REDLINE OF TMs UTILIZED
- NEARLY ALL CURRENTLY KNOWN SW & FW FIXES INCORPORATED
- 2ND ITERATION OF EARLIER SW & FW FIXES INCORPORATED
- $\geq$ 1 UYK-44 NCS UTILIZED

FIELD TEST #4

- ALL CURRENTLY KNOWN SW & FW FIXES INCORPORATED
- ITERATION OF EARLIER FIXES
- IC NEEDLINE INTERMEDIATE PERFORMANCE DEMO

FIELD TEST #5

- RESERVED FOR FINAL ITERATION OF FIXES
- ALL UYK-44 NCSs
- DRY RUN OF FORMAL PSV DEMO

Figure 7 PSV Field Test Objectives

weeks. The schedule for the PSV program is shown in Figure 8 [Ref. 30]. This chart also shows the schedule for the conversion of four NCS's to be upgraded from the UYK-20 to the UYK-44 computer suite. This was a corrective step in fixing NCS problems. Also listed are the two critical demonstrations (demo) required in the PSV program. The intercommunity needline demo was required to verify the proper performance of one NCS operating in concert with a second, as in a two brigade scenario. This was one of the two most critical deficiencies noted in TT-II. The second demonstration was the formal PSV system demo. This demo was based on formal Government technical test guidelines. Its plans required approval from the Government.

While the ultimate purpose of this demo was to verify system performance, a major focus was the second of the two critical deficiencies from TT-II: NCS reliability.

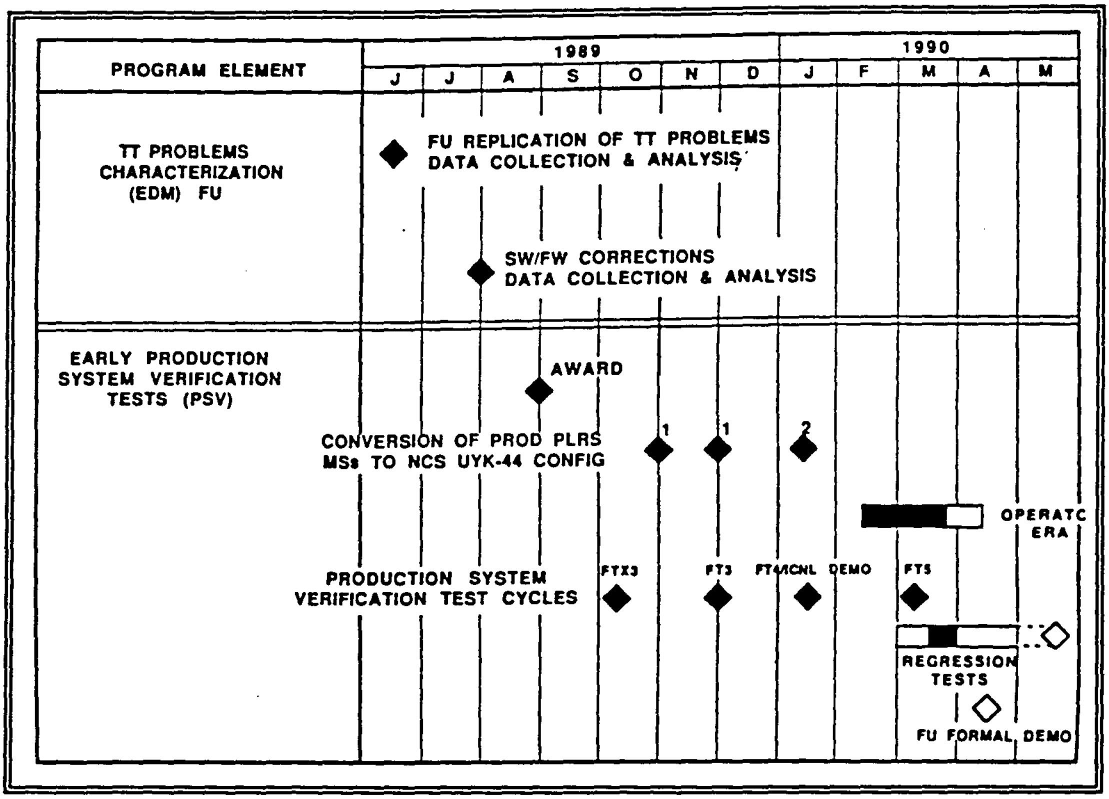

Figure 8 Production System Verification (PSV) Plan

3. Contractual and Logistical Considerations

The first two field tests in the PSV strategy were actually funded under the dollars remaining in the current contract. This was negotiated and agreed to by the EPLRS PM, the Government contracting officer, and the contractor. Because of this available funding, Hughes was able to start corrective action soon after the end of TT-II. Committed to salvaging the program, Hughes offered to perform the rest of the PSV plan on a pure cost reimbursement contract [Ref. 31]. The Government chose to use a fixed price type contract. Contract requirements in the statement of work reflected the inputs from the PM ADDS' restructuring plan, the TSM's new test requirements, and the Program Deviation Report Review Panel's conclusions referred to in the last section.

The logistical support for PSV was a large undertaking for Hughes. Field tests required manning of up to 84 stationary and mobile equipment sites. Hughes had to temporarily hire workers to man many of these sites. Other logistical support requirements fell into the areas of transportation, communication, maintenance, and training. Administrative functions, such as daily briefings and debriefings, worker problems, and personnel and equipment accountability were also required in the PSV plan. Logistical support alone was a major planning effort and consumed its share of the manpower and resources required to conduct PSV.

4. Summary

The focus of the PSV plan was limited to the specific problems identified in TT-II. It was resource intensive and involved many of the Government agencies involved in the previous technical test. The compressed schedule and iterative software and firmware correction cycle would require quality configuration management. To achieve success, the PSV plan would have to conduct two successful demonstrations. Failure in either of these demonstrations could be "fatal" to the EPLRS program, at least for the current contractor.

D. KEY GOVERNMENT ACTIONS

Software and firmware corrections accounted for the majority of the PSV plan. The difficult challenge to the Government was to manage the contractors execution of this software and firmware intensive corrective action while minimizing its own risk. The Government accomplished this with management actions that allowed them the visibility and control necessary for successful completion of the PSV plan. This section will detail these key Government actions. While the list is not all inclusive, it represents the critical actions identified during the research.

1. Software Engineering Assessment

An early step for the Government was to determine whether Hughes had the capability to correct the flaws in the EPLRS system. To accomplish this, CECOM directed its Center for Software Engineering (CSE) to conduct an assessment of Hughes' software process capability and the development of software in EPLRS. A team from the CSE conducted the assessment using the SEI's Software Capability Maturity Model. This was accomplished in July 1989. Hughes' software process was rated level II (above the 80th percentile of corporations) and the firmware process was rated level I-II (above the 50th percentile of corporations). The CSE team concluded that Hughes had the capability to isolate EPLRS system problems and successfully modify and test software and firmware to implement solutions. [Ref. 32] The Program Deviation Report Review Panel used this evaluation in determining its recommendation to the Acquisition Executive.

2. Contractual Considerations

Two important aspects of the PSV contract protected the cost exposure of the Government and provided strong control measures. The first aspect was to require specific demonstrations to show progress of the PSV plan. These functional and system demonstrations acted as milestones or progress points by which progress could be measured. The second aspect was a set of special provision clauses that contractually tied future program events and progress payments to the success of these demonstrations. Figure 9, below, shows how this worked. The dashed arrows connect the specific demonstration or test that must be completed to the event that will be initiated.

Additionally, the contract was written to ensure payments for the PSV system demonstration, considered a program milestone, would not be made until successful completion of the demonstration was recorded in the Government test report.[Ref. 33]

3. System Maturity Matrix

As described above, several future program events for EPLRS were dependent on demonstrating the maturity of the system. At the direction of the Program Deviation Report Review Panel, the EPLRS PM developed a system maturity matrix. The purpose of the matrix was to was to track the maturity of specific software and hardware technical parameters. The matrix showed the relation of the critical technical parameters to specific demonstrations and tests and the program decisions they supported. This matrix was a part of the TEMP and the program baseline. An example page of an EPLRS maturity matrix is shown in Figure 10 [Ref. 20:p. 4].

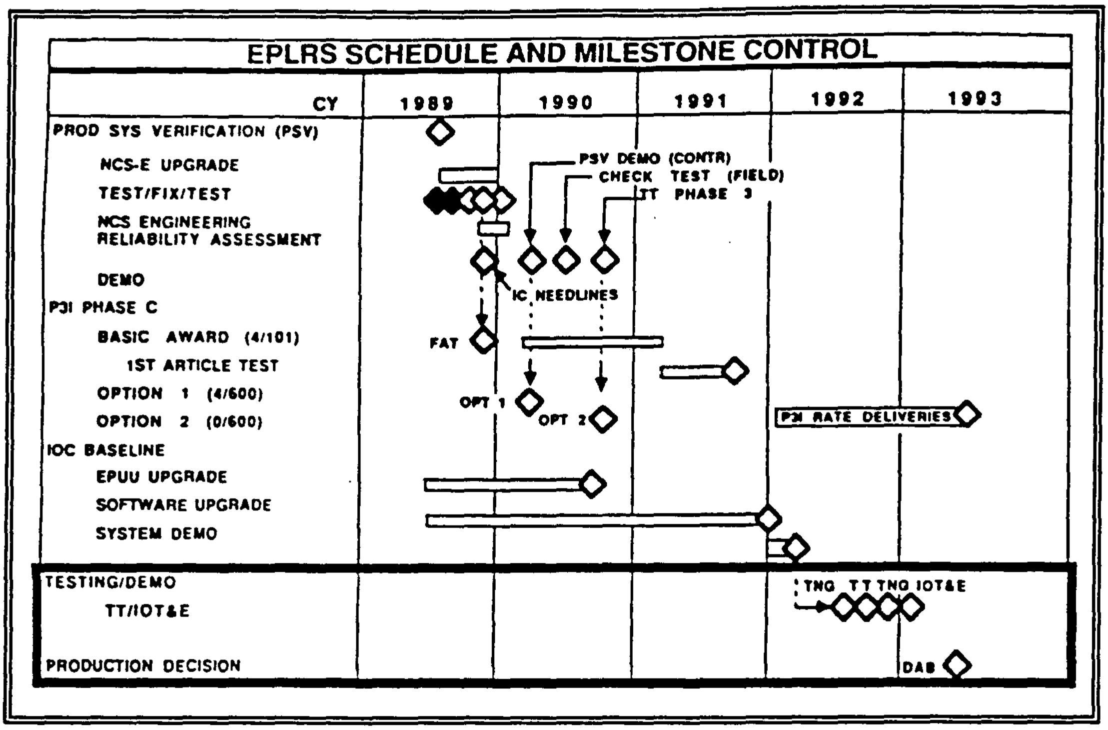

Figure 9 EPLRS Milestone Control Triggers

4. The California Field Office (CFO)

The CFO was the on-site representative of the EPLRS Government PM office located at the Hughes Fullerton plant. As described earlier, the CFO was a combination of Government and contractor employees responsible for the daily oversight of the program. During PSV the software and firmware related tasks of the CFO increased. Likewise, the staffing of the CFO increased from 15 personnel to 23 personnel. The number of contractors represented on the CFO increased from four to eight. The additions were all software and firmware technicians that assumed roles in monitoring field testing, system integration testing, and engineering of the EPLRS software and firmware.

Critical Technical Parameters	Total Events	Technical Objective and Threshold for each test event	Location	Schedule	Decision Supported	Demonstrated Value
Reliability NCS-E (1)	PSV FAT TT-III	80 hours MTBF 100 hours MTBF 125 hours MTBF	HAC HAC EPG	2Q FY 90 2Q FY 93 3Q FY 93	Award Opt 1 Go to TT-III Field System	79 hours
NCS-E(D) (2) Radio Set (2)	TT-III TT-III	172 hours MTBOMF 647 hours MTBOMF	EPG EPG	3Q FY 93 3Q FY 93	Field System Field System	
Availability NCS-E (2) Radio Set (2)	TT-III TT-III	.91 .94	EPG EPG	3Q FY 93 3Q FY 93	Field System Field System	.
Needline (1) Performance	PSV	75% (50/50)	HAC	2Q FY 90	Award Opt 1	96% (50/50)
	SMD	75% (90/90) 85% (80/80) 95% (50/50)	HAC	3Q FY 91	Award Opt 2	94% (90/90) 98% (80/80) 99% (50/50)
	FAT	85% (90/90) 95% (50/50)	HAC	1Q FY 93	Go to TT-III	
	TT-III	85% (90/90) 70% (90/90) (ECM)	EPG	3Q FY 93	Field System	
Safety (3)	TT-III	No critical safety issues	EPG	3Q FY 93	Field System	
OTAR (4)	TT-III	Performs as intended	EPG	3Q FY 93	Field System	

Figure 10 Sample Page of EPLRS Maturity Matrix

Testing greatly increased during PSV and so did the Government role in monitoring the software and firmware activities. The CFO staff conducted full time monitoring of the PSV field tests from set-up, to test, to post test analysis. The CFO also increased its role in monitoring regression and lab testing.

5. Use of Independent Verification and Validation (IV&V)

During the PSV plan changes were made in the IV&V structure for the program. First, an IV&V team was stationed on-site at the contractor facility as part of the California Field office of the EPLRS PM. The on-site IV&V team provided full time monitoring of the software work and testing and quick feedback to the EPLRS PM at Fort Monmouth. At the start of PSV, IV&V for EPLRS was conducted by two CECOM matrix organizations and the Marine Corps Tactical Software Support Agency

(MCTSSA). This complicated the IV&V effort. As a result, during the PSV plan, MCTSSA was given sole responsibility for EPLRS IV&V. MCTSSA was already performing software support for the Marine Corps' PLRS system. Their experience afforded the EPLRS PM detailed advice and comprehensive oversight of the contractor's software process.

6. Information Reporting

In addition to the monthly C/SCSC reports, the EPLRS PM had a full time information source through the CFO. The CFO was able to consult with the main PM office on important issues as they happened and relay status of the program whenever required by the PM.

During PSV the Government program office began using more metrics to measure program progress. Particularly helpful in this area was the tracking of Hughes' rigorous Program Trouble Report (PTR) system. PTRs were tracked by the number expected, actually initiated, corrected, and still open. The increased use of metrics afforded the PM greater visibility of the program status.

The Government and Hughes also began conducting "Tech Interchanges" every two months. These allowed discussion and work on program technical issues between the off-site Government representatives and the contractor representatives. This allowed for Government-contractor interaction on a more frequent basis than the standard IPR every six months.

Additionally, the EPLRS PM and his Hughes counterpart had an agreement to share information from the Hughes PMO meetings. The Hughes PMO agreed to format the briefing slides from their regular, weekly EPLRS meetings in a C/SCSC style format. Copies of the slides and other information from these meetings were sent to the EPLRS Government PM on a weekly basis. Hughes was very open about sharing information with the Government PM and kept the Government well informed.

7. User Involvement

TRADOC provided customer representatives at the contractor site to increase user involvement in the revision of the EPLRS system. Several of the deficiencies to be corrected were MANPRINT issues, such as lack of user friendly software in the NCS software programs. These U.S. Army personnel from the user community were trained on the EPLRS equipment and participated in the field tests at the Fullerton plant. They were debriefed after each test for comments on problems they experienced using the system and to gain their perspective on what they liked and disliked about EPLRS.

8. Testing

The scope of testing for PSV was greatly increased from that of previous work on the EPLRS contract. The statement of work for PSV called for three 2-3 day test cycles for the Test-Analyze-And-Fix portion of the plan. These were to occur approximately every six weeks. It also required a system level regression and verification lab test, an Intercommunity Needline field demonstration, and a PSV system field demonstration be conducted. All TAAF field tests and the two field demonstrations were to be patterned after the two brigade deployment scenarios used during TT-II.

E. KEY CONTRACTOR ACTIONS

PSV also posed a difficult challenge to the Hughes Aircraft Company, but a different challenge than that faced by the Government. The Government had to take actions that allowed it to effectively manage the contractor in his work. Hughes had to take management actions that facilitated the efficient and effective corrections of a software and firmware intensive system. This section will describe some of the key actions taken by Hughes to manage its work during PSV.

1. Contractor Openness and Cooperation

Throughout execution of the PSV contract Hughes maintained an open and cooperative attitude in its work with the Government. The Hughes PMO worked closely with the Government PM to develop the PSV concept and implement the plan. There was very open communication between the Government and Hughes PM's and they had a good working relationship. The Hughes PMO invited Government representatives to attend the weekly program reviews and the daily coordination meetings, and mailed the briefing charts of each meeting to the Government PM in Fort Monmouth. This cooperative attitude was apparent throughout Hughes' EPLRS program structure.

2. Exhaustive Test Strategy

Testing was listed earlier as a key Government action, but it was also a key Hughes action. Hughes developed the PSV plan with the Government and saw the need for extensive testing in order to mature the software and firmware in the EPLRS program. The testing was comprehensive and designed to produce the large volumes of data that, after being reduced, could be analyzed and used to fix problems. Figure 11 shows the progress made in solving EPLRS software and firmware problems with each iteration of testing.

The testing also required tremendous logistical support from Hughes. For example, Hughes had to hire temporary help to man all the test sites during much of the field testing. Hughes made the effort to hire primarily people who had previously worked for Hughes in a similar capacity. This added a degree of experience to the testing group.

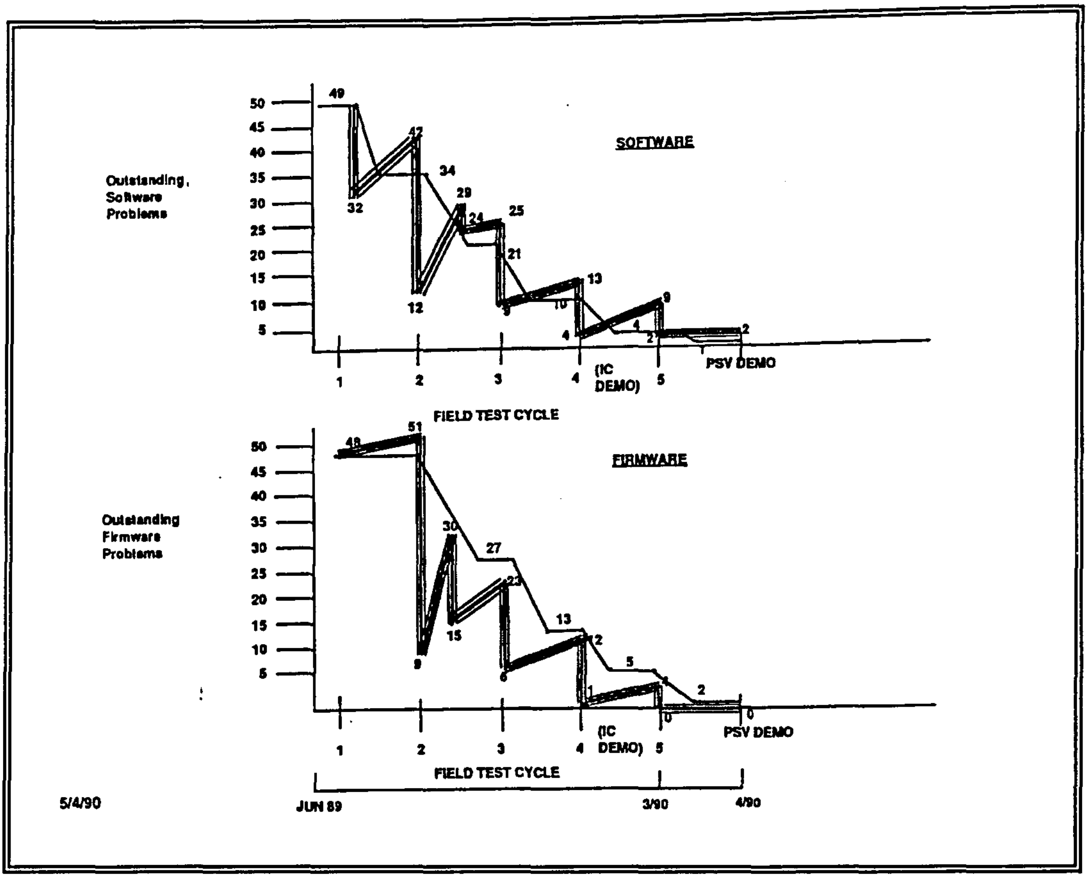

Figure 11 PSV Software/Firmware Problem Closure Summary

3. Team Concept

The Hughes Ground Systems Group is organized into functional divisions, such as its Software Engineering Division (SED). Support for the Hughes program offices comes from these divisions in a matrix type system. Yet, the support for the PSV program was organized under a team concept where the functional support personnel from the various divisions collocated with the PSV program management office. The PSV team included specialists from all functional areas critical to the program. The team concept provided the Hughes PM with increased access to the functional support required and created a more focused support relationship.

4. Corporate Involvement

The corporate leadership at Hughes was actively involved in the PSV program. Corporate leadership lent personal support to the plan and participated personally in various briefings to the D.A. staff. The PSV program office was given high priority for resources by Hughes corporate staff, to include selection of personnel for the program office and priority on use of the test bed assets for PSV requirements. The high level staffs also monitored the status of PSV closely. The Hughes PM for PSV attended weekly reviews with the division manager, monthly reviews with the group management staff, and periodic reviews with the Hughes corporate staff. PSV was a high visibility program at Hughes Aircraft Company.

5. Risk Management and Reduction

The test intensive nature of the PSV plan was a function of the risk management focus at Hughes. Risk management and reduction efforts were not properly emphasized in earlier phases of the EPLRS program, as evidenced by the results of TT-II. In the PSV plan, risk management and risk reduction were emphasized by both Hughes and the Government, and actually characterized the program. Planned events in PSV, such as its iterative testing cycle, frequent reviews, planned incremental improvements, and contractual milestone requirements, showed a strong emphasis on risk management and reduction from the perspectives of both Hughes and the Government.

F. RESULTS OF PSV

The PSV program was formally completed in May 1990. Its success had several measures. On 22 May 1990, the PEO COMM and the EPLRS PM briefed the DUSA (OR) on the successful results of the PSV system demonstration. The results of the demonstration were confirmed by both the Army Materiel Systems Analysis Activity (AMSAA) and the Operational Test and Evaluation Agency (OTEA). Accordingly, the Secretary of the Army certified that the system met PSV exit criteria. [Ref. 34]

The EPLRS system met or exceeded the exit criteria in every problem area during the required demonstrations. Consequently, option 1 of the LRIP contract was awarded to Hughes, as stated in the PSV contract. Additionally, an earlier requirement for a TT-II, Phase 3 was deleted from the EPLRS program test strategy by the DUSA (OR). An amended test schedule was approved for the EPLRS program that facilitated completion of system enhancements and progress towards IOT&E and a full scale production decision.

Hughes achieved other positive results from PSV. Several of the successful aspects of the PSV program have been carried over to follow-on program work. These actions include a more aggressive test approach, earlier and frequent integration of Government instrumentation and test equipment, and use of the team concept [Ref. 22:p. 2].

Since PSV the EPLRS system has successfully passed a third technical test (TT-III) and, at the time of this research, is preparing for IOT&E in April-May 1994.

G. SUMMARY

This chapter has attempted to accurately cover the events that guided the planning of the EPLRS PSV program and the execution of the PSV plan itself. The seriousness of the problems encountered by the EPLRS program in TT-II brought high visibility to the program. The resulting involvement by the D.A. staff and other agencies external to PEO COMM produced high level guidance that helped define the PSV plan. The actual plan was a discreet, comprehensive correctional plan committed to fixing the problems that had surfaced in the EPLRS system. Prudent actions taken by both the Government and the contractor properly addressed risk management and ensured success of the PSV phase of the EPLRS program. The PSV program did achieve success in meeting or exceeding all exit criteria and the EPLRS program was able to continue towards its goal of full scale production.

The next chapter will provide an analysis of the events described in this chapter. After analyzing the PSV program events, lessons learned will be extracted from the work.

VI. ANALYSIS AND LESSONS LEARNED

A. INTRODUCTION

By the measurements of success described in the last chapter, the PSV program was successful in correcting the EPLRS software and firmware shortfalls that surfaced during TT-II. This chapter will analyze the events occurring and decisions made during the PSV program in the context of corrective software management. The goal of the analysis is to link specific management actions of the Government and the contractor to the successful completion of the PSV program. Following the analysis will be a discussion of the applicability of this case to other DOD programs. This will be followed by a list of lessons learned generalized from the EPLRS case for application to other software intensive DOD programs.

B. ANALYSIS

This analysis will be structured around the key issues of the PSV program identified during the research. The events analyzed will include some that preceded the PSV program, but were instrumental in its design.

1. Involvement by Higher Level Authorities

The level of oversight involvement in EPLRS software corrective action was a function of the severity of the problem the program was facing. EPLRS' performance during TT-II resulted in both cost and schedule breaches of the program baseline. This brought a high level of visibility to the program and a mandatory investigation by a Program Deviation Report Review Panel. As head of this panel, the Deputy Under Secretary of the Army (Operations Research) (DUSA (OR)) became significantly involved in program oversight as the Army Acquisition Executive's representative. Concern by the DUSA (OR) and the TSM over the magnitude of the software corrective action required placed the program at great risk. Their support for the EPLRS program was critical for its survival. The serious software problems encountered by such a complex software system tested that support.

As EPLRS was a product of a larger project, PM EPLRS had two other higher level authorities: PM ADDS and PEO COMM. Both of these positions were involved in the daily actions of the EPLRS program and had a clear understanding of the situation. The resulting guidance from all of these positions formed clear boundaries within which the EPLRS program had to work to correct its problems.

This involvement added constraints and increased external pressure on the Government and contractor PMs in their effort to correct system problems. The added pressure and increased constraints were indicators of the loss of confidence by the higher level authorities in the program. Failing the expectations of the DUSA (OR) or the TSM in correcting the system deficiencies may likely have been fatal to the EPLRS program, resulting in its termination.

The resulting plan and thresholds for correcting the software and firmware in the PSV program closely followed the input from these higher level authorities. Adherence to this guidance and success in the program helped the Government and contractor program offices regain the confidence and maintain support of higher level authorities. The PEO COMM, PM ADDS, and PM EPLRS understood the importance of such support. They took steps to ensure the user community and DOD, as well as the right Congressional staff, understood the PSV program. Key players were given briefed in order to build support for the program. [Ref. 35]

Maintaining this confidence and support was critical to conducting the corrective action plan. Without the confidence and support of the TSM and the decision making authorities in the program's chain of responsibility the EPLRS system would have had little chance of survival.

2. Risk Management

Several actions that were either precursors of PSV or actual steps in the plan were indicative of good risk management.

The plan to restructure the program showed an appreciation for the high level of risk involved in developing unprecedented software by the original program strategy. The five goals of the restructuring plan (see Chapter V, section B.1.) were each explicitly focused on reducing a source of risk related to the system's software problems. While this restructuring plan affected the entire EPLRS acquisition strategy, it had a marked influence on the formulation of the PSV plan itself. The restructuring plan removed the concurrency in the EPLRS acquisition strategy in an effort to shift to a more conservative developmental approach. This approach led to the PM ADDS' proposal to ensure a clear separation between the PSV phase of the EPLRS effort and a concurrent proposal to modernize the NCS. The Program Deviation Report Review Panel supported this position with this statement: "...the panel holds the view that the first step in 'getting well' must be directed to achieving a successful system performance demonstration unencumbered by a simultaneous attempt at NCS modernization" [Ref. 27]. The plan to fix the software and firmware problems of EPLRS was in itself a complex task. This effort to simplify and focus the software corrective action by isolating it from other program tasks reduced the risks of configuration problems, competing resources, and integration difficulties, to name a few.

The restructuring plan moved the program towards more of an evolutionary acquisition strategy. An evolutionary acquisition strategy is one that produces a basic model to meet initial operating capability (IOC) and incrementally builds improvements to that model [Ref. 36]. Just as with the system as a whole, an initial software functionality would be developed and delivered. This strategy would implement the essential software corrections to field a lower risk, IOC model. Developmental risk is reduced because such factors as the size of the software development team and the possible communications links are reduced in comparison to a grand design project. Because the size and degree of functionality of the IOC software is reduced, development complexity is lower. New software builds and planned system improvements of higher risk would be included in block upgrades for EPLRS after its initial fielding [Ref. 28]. This incremental strategy also allows better focus on a smaller scope of work. This reduces risk in design and testing of the program.

The restructuring plan also addressed risk management techniques involving testing and contractually protecting Government cost exposure. These actions will be addressed individually later in this section. Including these actions, the influence of the restructuring plan was clearly evident in the resulting PSV plan. It strongly emphasized reduction of risk involved with the software process for the corrective action necessary in PSV and in follow-on program phases.

Another risk management action taken by the Government was to conduct an independent technical assessment of the software capability of the contractor. The purpose was to assure the Government that the software capabilities existed at Hughes to resolve the problems from TT-II. The results of the assessment were that Hughes had a level 2 software capability and a level 1.5 firmware capability as measured by the SEI capability maturity model. There were two positive results. First, the assessment convinced the Government that Hughes had a more than adequate software development capability to perform the needed work in PSV. This acted to reduce the risk in the decision to stay with Hughes as the prime contractor. It also increased the Government's confidence in the ability of Hughes to perform quality software work. Secondly, while the assessment described a disciplined and repeatable software engineering process at Hughes' SED, it pointed out some weaknesses in their firmware engineering process. Hughes used these results and recommendations to improve their firmware development process. Because EPLRS is firmware intensive, this proved beneficial to both EPLRS and the PSV program.

The Government also managed risk through contractual actions. The use of special provision clauses in the PSV contract reduced liability of the Government during the execution of the contract. These special provision clauses had two important effects.

First, the clauses stated that specific future program events, such as first article test (FAT) and options 1 and 2 to the current LRIP contract, were to be exercised only upon successful completion of a required demonstration or test. Each of these future events was contractually tied to one of these demonstrations or tests. Secondly, special provision clauses held the contractor monetarily liable for system shortfalls during the two field demonstrations and technical test II phase 3. Failure at any of these tests would require the contractor to conduct analysis and make corrections as required to pass the event to Government satisfaction, at no change to the fixed price contract. These same clauses also allowed the Government contracting officer to stop progress payments for any of these test failures. The special provisions section of the PSV contract modification is included in appendix 1. [Ref. 33]

These contract clauses served to reduce the risk of the Government by reducing future program liabilities until Hughes passed specific milestones. The clauses increased the monetary risk to Hughes, motivating Hughes corporate to place a high priority on the PSV program. Because Hughes had lost some of the Government's confidence after failing TT-II the situation warranted a greater shift in contract risk from the Government to the contractor.

The last action to be discussed in this section is the PSV maturity matrix. One conclusion of the Program Deviation Report Review Panel was that the EPLRS PM must develop a maturity matrix and include it in the EPLRS TEMP and program baseline. This matrix would assist the Government and the contractor in tracking the contractual requirements described above. It clearly listed the required software technical parameters to be met and related them to the demonstrations or tests they occurred in and the program decision they supported. This matrix provided a tool for accurately tracking the maturity of the system. It used that maturity as a measurement of risk.

These actions on the part of the Government highlighted the effort to manage risk in the PSV plan. Shifting to a more conservative strategy and isolating the corrective action plan from other work helped to better address the risk inherent in correcting software problems. The independent software process assessment and use of special contract provisions were actions that further reduced the Government's risk. Based on the high level of risk involved in the EPLRS software corrective action plan these were prudent actions.

3. Testing

One factor contributing to the TT-II failure was an underestimation of system testing required for the engineering development model (EDM) of EPLRS [Ref. 22:p. 4]. The test-analyze-and-fix (TAAF) methodology adopted for PSV provided a comprehensive, iterative test strategy that incrementally identified problems and corrected them for the system. This type of test methodology was well suited to address the number of problems present in the complex software and firmware of the EPLRS system. The resulting test plan for PSV had two important characteristics that differed from previous contractor testing for EPLRS. It focused on field testing (vs. lab testing) and used a far more realistic test environment. Hughes had learned that failing to stress the software and equipment under more realistic field conditions, rather than simulation programs, masked many of the software faults in the system. This was partly responsible for some of the software immaturity in TT-II.

Additionally, Hughes adopted the practice of conducting detailed dry runs before each demonstration or test. These "dress rehearsals" represented the actual event and were witnessed by Government program personnel. Conducting dry runs better prepared Hughes for each event and reduced the risk involved.

Design of the test strategy was strongly influenced by lessons learned from previous testing. Based on the complexity of the software and the risk of the program, the iterative TAAF methodology was an effective choice for the PSV program.

The Government required two demonstrations be integrated into the test plan to serve as milestones or progress points. These two events required the demonstration of performance parameters that were listed as critical shortfalls in TT-II. The second demonstration was also used to demonstrate total system performance. The Government and Hughes were able to use these demonstrations to show the progress of the PSV program and to verify the fixes of noted shortfalls from TT-II. The demonstrations served as easily quantifiable benchmarks to track the readiness of the program to enter the next phase of Government testing.

PSV also had a significant increase in the amount of participation by Government workers. Testing, a major portion of the PSV plan, benefitted from this. Government personnel from the CFO advised the contractor with preparation of test documentation prior to formal submittal. This assisted the contractor in producing test documentation that would be more acceptable by the Government. After submitting the documents, face to face meetings with Army technical representatives helped identify any misunderstandings in the test concept or plan. It also identified any problems prior to the actual testing. This cooperation before the actual testing benefitted Hughes. By closely scrutinizing the test plans with the Government, Hughes would have nothing to hide at test time. Test problems would then not be viewed with suspicion by the Government. The Government also had a more accurate base of knowledge to assess the test progress from and could better assist Hughes in problem solving when necessary.

Army customer representatives worked in-plant with Hughes engineers to improve requirements interpretations. This allowed Hughes to verify their interpretation of the software and system requirements before testing. These users also participated in field testing. Assessments of equipment operations from the users helped to solve the MANPRINT problems of the system. This user input added expertise that was not widely used, nor available, prior to PSV and helped make the software more user friendly.

Personnel from the Army test community also assisted Hughes on-site in conduct of the PSV testing. EPG personnel and their contractor responsible for the test instrumentation worked with Hughes engineers to ensure their test equipment interfaced with the EPLRS system. They also assisted with the software that ran the test scenarios and the data reduction requirements after test. The EPG personnel also assisted Hughes in making test planning representative of the testing requirements at EPG. This expertise increased the effectiveness of the Hughes PSV program as a preparation for the next phase of Government testing.

Lastly, Hughes made significant improvements in their test simulation and analysis support tools for the PSV program. This equipment helped to improve monitoring and data collection during testing. The simulation tools provided a much more realistic test scenario in which to adequately test and stress the software in the EPLRS system. The lack of such test equipment in earlier testing masked a number of problems and decreased the efficiency of the test efforts. The improved equipment even allowed for better test planning. Since scenarios were more accurate, the time required for adequate data collection was better known.

Overall, the test strategy, plan, and assets were greatly improved over previous tests' efforts. The improved test capability helped to ensure a more effective effort in testing the software and verifying its proper correction. Hughes' efforts to conduct exhaustive field tests in a realistic environment advanced the maturity of the software and firmware.

4. Organizational Structure

Both the Government and the contractor made adjustments in their organizational structure for the conduct of PSV. These changes were centered around moving software expertise closer to the center of program events.

The Government PM for EPLRS already had an established California Field Office at Hughes' Fullerton plant. At the initiation of PSV the PM increased the number of software technicians in the CFO. Their role was to closely monitor and assist all software and firmware functions by the contractor throughout the PSV program. Their presence provided both the Government and the contractor with a valuable source of advice on software and firmware. The contractor could gain advice on Government requirements and standards for the software work being performed. The Government PM could receive daily feedback on software corrections and new issues. With software technicians acting as the PM's "eyes and ears" at the contractor's plant, the PM was always knowledgeable about the software status of the program. Awareness of this status was critical for the PM.

An important group in the CFO was the IV&V team. Originally the IV&V role was filled by three different agencies: two CECOM matrix support organizations and MCTSSA. During the PSV program MCTSSA was selected to be the single IV&V agent for EPLRS. This provided a number of benefits for the program. First, going to a single IV&V agent provided better control of the IV&V function. IV&V work was not duplicated and the single agency reduced confusion. Secondly, MCTSSA had the most experience in the system. MCTSSA was the Marine Corps' software IV&V agent,as well as post deployment software support agent, for the PLRS program. As such it already had a great deal of experience with the CMS2 program language, the language used in EPLRS at the time. Also, over 50 percent of the software code between PLRS and EPLRS was common to both systems [Ref. 36]. This made the choice to select MCTSSA as the single IV&V agent a logical move beneficial to both the Government and contractor PM. Another important consideration in the decision was that MCTSSA was well suited to assume the PDSS role in the future. As previously stated, they performed this function for the PLRS program and already had the assets required to conduct the work. The decisions to work with MCTSSA in these roles married the EPLRS program with the Government agent perhaps having the most experience in this system. It proved to be a valuable source of expertise to the program.

Hughes also made adjustments to its organizational structure by using the team concept to staff the PSV program office. The "Tiger Team" concept placed Hughes support personnel under the complete control of the PSV PM for Hughes [Ref. 29]. This reduced the administrative restrictions of the normal matrix type functional support for program managers at Hughes. The PSV PM now had consistent manning in the technical support positions of the office and better control and influence over the staff. This gave the PM dedicated software and firmware technicians that reported directly to the PM rather than the SED.

Changes in organizational structure were made by both the Government and the contractor in the PSV program. These changes allowed program offices to put software and firmware technicians at the critical spot where they could best impact the software and firmware corrective action. Both the on-site Government representatives and Hughes' "Tiger Team" formation provided software expertise to focus on the software issues. It place more control in the hands of the program managers.

5. Information Flow

Communications between the Government EPLRS PM and the Hughes PM was frequent, open, and most often conducted through informal channels. While more formal channels of communication were already in place, such as the C/SCSC reporting system or IPRs every six months, the informal channels often provided a more valuable and timely source of information for accurate program information and decision making.

One of the key sources of information for the Government PM was the CFO. The on-site presence of the CFO offered the PM immediate Government status on software test results and other software issues during the corrective action process. The CFO also helped in the relay of information between the PMs. This immediate source of information facilitated decision making for both the Government and contractor PMs.

Another source of Government information was directly from the contractor. During this corrective action effort the desire for open lines of communication was mutual for both PM offices. The Hughes PM agreed to provide as much program information as possible to the Government PM. For instance, Hughes' briefing charts and minutes from the weekly contractor program status meetings were immediately mailed to the Government PM. Hughes' program personnel made frequent use of Electronic mail and facsimile communications with the Government PM office. These channels were used to keep the Government office continually updated on the status of the software corrective action. [Ref. 36]

"Tech interchanges" were another non-standard channel of information. These face to face meetings allowed more frequent exchanges between the Government and contractor counterparts than did the semi-annual IPRs. This improved the technical coordination for the program.

The extensive use of these informal channels of information by both the Government and the contractor facilitated the corrective action plan. Accurate and timely knowledge of program status assisted the Government PM in decision making and communication with oversight agencies. The shared knowledge between the Government and contractor PMs made their communications more effective. Faster insight into program problems allowed solutions to be developed more quickly. The nature of the corrective action plan required a frequent source of open communication between the program offices.

6. Government-Contractor Relations

The PSV program was characterized by a good working relationship between the Government and Hughes. Hughes maintained an open and cooperative attitude throughout the length of the program. Hughes was involved up to the corporate level in the design of the PSV plan. Their managers participated in briefings to various Government agencies during the planning process of PSV. These actions showed the Government a sense of dedication and commitment from Hughes to succeed in correcting EPLRS problems.

In particular, a high level of cooperation existed between the Government EPLRS PM and the Hughes PSV PM. As mentioned above, Hughes openly shared program information with the Government. They invited CFO personnel to attend all of their program office meetings. Briefing charts from the Hughes meetings were prepared in a format that would be most helpful to the Government PM office. These actions were above those called for in the contract. This open and cooperative attitude built a sense of trust between the counterpart PM offices and created a productive environment.

Another significant indicator of Hughes' cooperation was the level of involvement by Hughes' corporate leaders. In both the planning and execution of the PSV program Hughes' corporate leaders were actively involved. They helped brief

Government decision makers, provided input to the planning process, and pledged full support of the corrective action plan in terms of Hughes resources. The corporate staff gave the PSV program full priority for personnel and industrial assets at the expense of other Hughes' programs. This corporate involvement by Hughes served to both facilitate the work of the PSV program office as well as satisfy Government officials of the commitment of the corporation to the correction of the EPLRS system.

The good relation between the Government and Hughes was a significant factor in the success of the PSV program. Adverse Government-contractor relations, not uncommon in stressful program situations where corrective action is required, often hinder the effectiveness of program offices in managing work. Lack of support at the corporate level can constrain the efforts of a program, impeding chances of program success. Fortunately for the EPLRS program this was not the case.

C. GENERALIZING THE EPLRS CASE TO OTHER PROGRAMS

Obviously, every program situation is unique. Among software intensive programs the potential problems that may be encountered are numerous, each presenting a different challenge. But, there are some common threads in the requirements of software intensive programs with problems that warrant corrective action. This section will first address the unique characteristics of the EPLRS corrective action case. It will then discuss those characteristics of the EPLRS case that may be viewed as common and useful to other software intensive DOD system programs.

1. Unique Characteristics of the EPLRS Case

In this discussion the term "unique" refers to those characteristics of the EPLRS case that may not be easily generalized to other DOD systems. These characteristics often required unique methods and techniques for dealing with the EPLRS program. An assumption that is made assumes that the methods and techniques referred to would not readily apply to most other programs.

The EPLRS system evolved from the recently developed Marine Corps system, PLRS. This resulted in the system having some previously developed components and software. The computer language, CMS2, and the system expertise for this type system were familiar products to the Marine Corps and Navy. As a result, this fact drove many of the software support decisions, such as the selection of an IV&V agent and PDSS organization.

Because EPLRS was an enhancement from an already developed system, the complexity of the remaining development was underestimated [Ref. 31]. This proved to be a driving force behind the accelerated schedule and level of concurrency in the original acquisition program strategy. It that led to system problems and also drove many decisions during corrective action. These included adopting a more conservative program acquisition strategy and developing better simulation and analysis tools. These needs had not been forecasted earlier.

EPLRS was also unique in that it was a communication "pipeline" for several battlefield data systems also currently under development. EPLRS was considered the "linchpin" for this high priority network of Army data systems grouped under ATCCS

[Ref. 28:p. 1]. This fact may have been responsible for the level of concern and oversight involvement the program saw after its poor performance in TT-II. The involvement and support of high level authorities may have afforded the EPLRS corrective action program a level of clout and resource priority not always available to other programs.

One other characteristic that may be viewed as unique is the fact that EPLRS had breached its baseline and went through a program restructuring. While restructuring can happen to other programs, it may warrant corrective actions that may not be feasible for programs that have not breached baseline. The level of effort and cost that resulted in the EPLRS corrective action plan would probably not be available to program that had software problems but maintained baseline integrity. The issue in this case is perhaps a question of scale.

These issues are those that are the most unique to the EPLRS program. They have had an effect on corrective action program decisions and may not apply to other programs. These characteristics will limit the applicability of lessons learned in the EPLRS case.

2. Common Characteristics of the EPLRS Case

Outside of the characteristics listed above, the EPLRS case has several issues in common with other software intensive programs that have experienced software development problems. The program entered formal Government technical testing with immature software that could not fully satisfy requirements. It is a system that relies heavily on software to accomplish its mission. Software was in the critical path of the system's development plan. EPLRS is required to interface with several other battlefield systems and has numerous human interface requirements. Perhaps most significantly, EPLRS was managed by Government and contractor program offices that were faced with serious decisions after identifying critical software shortfalls in their system. To see how common these characteristics really are one need only review the GAO reports that cover software problems in major defense weapon systems. Not withstanding the unique characteristics of the EPLRS system, many of the lessons learned from the EPLRS corrective action case can be applied to other programs in a reasonable manner.

D. APPLYING LESSONS LEARNED TO OTHER PROGRAMS

This section presents a list of corrective action lessons learned that were generalized from the EPLRS case analysis. Following the logic of the previous section, these lessons learned should be applicable to software intensive programs that encounter software development difficulties.

1. Ensure software corrective action plans are designed to meet the requirements and expectations of those parties that oversee the program. This refers to the user community and the chain of responsibility of the program.

2. Maintaining the confidence of the TSM and higher level authority throughout a corrective action plan is critical. Without the TSM and decision making authority having confidence that the plan can viably correct the software problems, the program has little chance of survival.

3. Corrective action in software intensive programs normally warrants a shift to a more conservative strategy. An incremental, deliberate approach can better address the risk inherent in this situation.

4. Corrective action on software in a system should be isolated from any simultaneous attempt to enhance or otherwise engineer changes into the software. The focus of the corrective action should only be the existing deficiencies.

5. If a contractor's software process capability is questionable request an independent technical assessment of that process. The SEI's software capability maturity model is a good format to use.

6. Ensure there is a contractor plan to incorporate recommendations from the technical assessment into the software and firmware process for the program.

7. Where appropriate, contractual clauses can be used to reduce Government risk during software corrective action. An example would be the EPLRS contract clauses stipulating that system demonstrations had to be successful before future contracts options would be awarded.

8. Corrective action requires iterative testing to confirm software maturity. This occurs as consecutive tests identify and correct more problems and the software reliability increases. The TAAF methodology provides an exhaustive testing plan commensurate with the task and level of risk normally presented by a program requiring corrective action.

9. Early Government involvement and support in corrective action better prepares contractors for follow-on Government testing. Government participation can reduce requirements confusion, improve user satisfaction, enhance communications, and provide Government test expertise to contractor test personnel.

10. The scope of contractor software testing must replicate the actual environments and scenarios to be encountered at Government testing.

11. Requiring interim demonstrations during the test cycle provides the Government PM with progress points to track program progress. Used in an event-based program strategy, these interim milestones can reduce the risk in making Government program decisions.

12. An on-site Government office at the contractor's facility gives the PM a significant advantage in managing corrective action of the software and firmware on a daily basis.

13. Put the agency most experienced in the system's software in the IV&V role for the program. The IV&V personnel must be able to see problems not detected by the developer. Strong IV&V and software system experience provides the intuitive

insight and appreciation for problem areas needed to effectively perform the IV&V role.

14. The team concept for contractors, where software personnel are assigned to the program office, is more effective than matrix type support.

15. Maintaining control of a corrective action plan requires frequent updates and communications. A reliable, informal communication system with the contractor is essential.

16. Open sharing of information between the Government and the contractor improves decision making and reduces surprises on both sides.

17. Government-contractor cooperation is critical to a smooth corrective action plan.

18. Corporate involvement is necessary to provide the priority support required of an intensive corrective action plan.

E. SUMMARY

No single action or "silver bullet" made the EPLRS corrective action plan, PSV, a success. Productive involvement by many key players combined to make PSV work. Careful analysis and planning designed a program that would effectively correct the system's software and firmware and properly manage the risk involved. Conservative and disciplined testing efforts and managerial control over the program overcame the difficulties of correcting software in such a complex system. These factors combined with the good working relationship of the Government and Hughes Aircraft Company were responsible for the success of the EPLRS corrective action.

While the EPLRS system has some unique characteristics, it has enough commonality with most other software intensive systems to be helpful as a case study.

The lessons derived from the EPLRS analysis should be applicable to many similar systems.

The next chapter will provide conclusions to the EPLRS case study. It will also provide a set of recommendations related to the lessons learned in this chapter.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. INTRODUCTION

This chapter will begin by presenting conclusions to the research effort. The conclusions will be followed by a set of general recommendations to be considered as possible ways to improve the acquisition of software intensive systems by DOD. The chapter will also include answers to the thesis questions used in this research effort. Recommendations for further study will be included at the end of the chapter.

B. CONCLUSIONS

The opening chapters of this thesis described the critical role software plays in today's defense weapon systems. The vast majority of these systems rely on complex software that is very challenging to develop and absolutely essential to each system's performance. The development of this software involves millions of source lines of code and costs billions of dollars annually. Additionally, it is a fact that high levels of risk and a high probability of encountering problems are inherent in today's software development environment. Recent estimates show that 7 out of 10 major weapon systems currently in development are encountering software problems [Ref. 11].

In this environment program offices of software intensive systems must be prepared to address problems they are likely to face in development. Ideally, programs should be proactively focused in an effort to avoid software developmental problems. Yet, the statistical probability of developmental problems occurring is relatively high. Therefore, an astute program office should be prepared to implement and manage a software corrective action plan. Likewise, the DOD environment should support and facilitate the efforts necessary to effectively correct the software system shortfalls.

The EPLRS case studied in this paper presents an example of successful management of a software corrective action plan. The planning and management of the PSV program by both the Government and Hughes Aircraft Company effectively corrected the problems with the EPLRS system. The program efforts were completed within given cost and schedule constraints and satisfied higher level authorities and the user community. The reason this corrective action plan succeeded was because it was well designed and effectively managed.

The sound decisions and prudent actions of both the Government and the contractor resulted in mature system software prepared for future Government testing. The corrective action strategy and management techniques used in PSV were responsible for minimizing Government risk, maintaining a sense of teamwork between the Government and the contractor, and increasing the chance of program survival. Overall, the PSV program presented a timely, effective, and efficient corrective action plan that allowed a program to successfully continue with its acquisition strategy.

The lessons learned in the EPLRS case can be applied to other software intensive systems in DOD that are experiencing software problems. There are some unique characteristics in the EPLRS case that set it apart from other programs. Yet, the vast majority of circumstances and learning points involved in the correction of EPLRS software and firmware are common to the DOD procurement environment. Therefore, the lessons learned can be readily generalized to a large population of software intensive systems being developed by DOD. The design concept and management techniques used in the EPLRS case may serve as models for the design and implementation of software corrective action required by other DOD weapon systems.

Lastly, this thesis considered the actions of a single case, the EPLRS program. Studying only a single case naturally reduces the external validity of the results. In the case of corrective software management in the EPLRS program, the concepts analyzed seem readily generalizable to a large population of DOD programs. While the conclusion is that the lessons from this case are applicable to many DOD programs, the limitations of the single case study should be noted.

C. RECOMMENDATIONS

The following recommendations are a result of this research.

1. Develop a DOD Policy on Management of Software Corrective Action

The lack of an effective software corrective action plan can prolong developmental problems in a weapon system and cause significant cost, schedule, and performance shortfalls. In many cases, software problems affecting the development of a weapon system may not be properly addressed in an effort to avoid visibility. Still other systems may have software problems for which the corrective action is underestimated. In the case of EPLRS, serious software problems were not identified until technical testing. Ideally, critical software problems would be identified earlier in the development process. A measurements program, such as the Army's STEP metrics, would assist in identifying such problems earlier.

A DOD policy is needed that encourages a consistent approach to effectively manage the corrective action of software problems. The policy should promote conservative methodologies, such as TAAF, and emphasize comprehensive risk management in the corrective action plans. It should address the subject of schedule readjustment and baseline breach. Guidance should be to complete corrective action before further system development involving software is attempted. A major intent of the policy should be to remove any potential threat PMs may perceive exists in expanding a program schedule to allow for the correction of software problems.

2. Develop a DOD Model for Software Corrective Action

A general model of a software corrective action plan would facilitate the implementation of corrective action by program offices. Different models may be developed for various categories of defense systems that benefit from a tailored software corrective action plan. These differences may involve testing requirements, availability of test assets, scheduling of corrective action events, etc.

The corrective action model should address identification of root cause and test methodology. While the methodology, as a rule, should be conservative, it should also be flexible enough to be tailored to a program's needs. The model should also address risk management through required interim progress demonstrations, comprehensive verification and regression testing, and involvement of Government users and testers to clarify requirements and verify test environments. The use of such models will also promote some standardization for Government and contractor program offices.

3. Use Separate Contractual Efforts for Corrective Action

When possible, software corrective actions should be conducted under separate contractual efforts (this includes contract modifications). This provides for a clearly defined scope of the required action. Its own statement of work and list of deliverables clearly define the tasks to be accomplished in the corrective action. The contract can be used to appropriately share the risk between the Government and the contractor. Special provision clauses can reduce fiscal exposure of the Government by stipulating successful demonstrations of required performance before payments or approval of follow-on contract events. A contract provides a tool necessary to control the corrective action for the Government.

4. Use of On-Site Program Office Personnel during Corrective Action

The use of on-site software technicians provided an indispensable asset for the PM during software corrective action. The level of monitoring of software testing and engineering significantly increases during corrective action plans. Additionally, meetings and reporting requirements increase. The use of on-site personnel is the most effective way to meet these requirements and keep the main Government program office updated on all activities. These on-site personnel also provide a useful liaison role for the contractor. In many cases, approvals, advice, and decisions can be made immediately by on-site personnel. This avoids delays from waiting on periodic visits from program office personnel or the inevitable delays that result from passing documentation back and forth between program management offices. This also facilitates the IV&V mission. On-site IV&V representatives can provide close, daily supervision of all software efforts. These personnel can accommodate the increased close monitoring necessary during corrective action of a systems' software.

D. ANSWERS TO THESIS QUESTIONS

1. How can a PM most effectively manage software corrective action to solve problems that develop in the software acquisition process?

A PM can approach corrective action required for software in a number of different ways and be successful. Each program situation presents a different set of requirements and constraints that corrective action must be tailored to. The EPLRS case shows one successful corrective action plan. The general concepts drawn from this case attempt to answer this thesis question. These concepts proved successful for the EPLRS program and can be applied to other programs as well. Applying these concepts to other programs needing software corrective action will assist in generating the actual tasks for that specific case.

The general concepts developed from studying the EPLRS corrective action case are as follows:

- Understand what the user community and decision authority want from the corrective action plan. Maintain their confidence.
- Establish a comprehensive risk management effort for the corrective action plan. This characterizes the plan itself.
- Isolate the corrective action effort from other program efforts. This simplifies the tasks, ensuring better error correction and configuration management.

- Keep the operational testers involved early in the corrective action plan.
- Use a test methodology that exhaustively tests the systems and each of its parameters. The test analyze and fix methodology works very well.
- Use field testing to accurately replicate the operational environment of the system.
- Keep the user involved in technical testing.
- Establish an on-site office for Government program management personnel. It improves the oversight role of the software technicians and the IV&V personnel.
- Work to establish and maintain good relations with the contractor. The corrective action effort is facilitated by teamwork between the Government and contractor program offices.
- Establish an open and frequent communications channel from the contractor site to the PM office. This information flow is critical.
- Look for corporate involvement from the contractor. It can provide significant support and priority to the contractor PM. In corrective action programs this is often essential for success.

2. How can the organizational structure of the contractor and Government program management offices facilitate the software corrective action plan?

Both the Government and the contractor can tailor their program offices to better fit the needs of a corrective action plan. The use of an on-site office by the Government PM can greatly increase the oversight capability and information flow to the home office. Increasing the software technicians in both Government and contractor offices is critical during this type of work. The increased test and analysis, often conducted at a faster pace than normal, will require more technicians to execute it and oversee it. This will also increase the IV&V role. The goal of the structural change should be to get the expertise and actions where they are needed most.

3. What is the PM's relationship with the contractor in correcting software deficiencies?

Higher risk and pressure are inherent in any corrective action program. An adverse relationship in this type of an environment can be detrimental to the program. Teamwork between the Government and contractor program offices is very important to implementing a corrective action plan. Open and cooperative relations help in problem solving and decision making between both parties. This is facilitated by agreeing to share all information between the program offices. Both the Government and contractor PMs have to fully and actively support the teamwork philosophy. An open and frequent information flow keeps both PMs better informed and reduces unwanted surprises. This relationship is further improved by positive and supportive corporate involvement. This involvement facilitates the corrective action and improves the confidence of the Government that the contractor is committed to the corrective action of the system.

4. How is risk management addressed by the PM office in a software corrective action plan?

As stated earlier, an effective corrective action plan is characterized by risk management. All major actions in the plan should be considered in the context of risk management.

The corrective action plan should have a narrow scope focused on clearly identifying the software problems and designing a plan to fix only those problems. The testing plan should be designed as a conservative, iterative effort that allows careful analysis and correction between tests. Contractual actions can be used to reduce Government risk and protect the Government's fiscal exposure through special provision clauses. Independent technical assessments of the contractor's software capability can be conducted if it is questioned. These are only a few of actions that can help manage risk. But risk management in corrective action is critical and must be strongly emphasized throughout the plan.

5. What information reporting system allows the PM to best monitor the progress and manage the actions of a corrective actions plan?

The information flow to the Government PM becomes even more important when conducting corrective action. Status is needed more often, reports to higher authority are often required more often, and critical decisions may be required more frequently. This requires an accurate and frequent flow of information between the contractor facility and the PM office. The standard information and reporting channels often do not satisfy this information requirement well. Agreements between the contractor and Government PM for the sharing of information and reports can facilitate this need. The sharing of information between the PMs is critical to decision making and is best implemented by agreed methods of communication. This may take the form of E-mail, fax messages, minutes to internal meetings, and sharing of internal reports, to name a few. The on-site Government office can also provide a consistent, daily flow of information and reports to the PM. This office can act as a liaison for gathering information from the contractor and providing the contractor the same from the Government PM.

6. How can contractual considerations be used to manage software corrective action?

The contract gives the Government PM a tool to better control the corrective action plan and effectively manage the risk. The contract action can be negotiated as a separate contract or a modification to a current contract. The scope and statement of work (SOW) provide a manner for the PM to clearly define the goals, constraints, boundaries, and criteria for the corrective action.

The contract can be used to shift risk from the Government to the contractor when warranted. Special provisions clauses can motivate the contractor by tieing the award of future program events to the success of system demonstrations. Progress payments can also be made dependent of successful demonstration of system performance.

E. RECOMMENDATIONS FOR FURTHER STUDY

1. DOD Policy on Correction of Software Problems

Examine the need for a DOD policy addressing the proper management of software development problems. The "software crisis", as it has been called, presents problems throughout the DOD procurement arena. The impact on cost, schedule and performance across DOD warrants policy that promotes an effective manner of dealing with these software problems. The study may consider the opinions of current PMs and PEOs service-wide, the opinions of contractors, the potential long range impact on cost and schedule, and what barriers may exist to such a policy.

2. Role of Contracts in Software Corrective Action

Examine how contractual efforts can be used by the Government to better manage software corrective action plans. The contractual actions for a software corrective action plan have a significant impact on the risk assumed by the Government. Contracts can be tailored to have a variety of effects. This research can examine risk sharing between the Government and the contractor. It may also study the role of contract incentives, and the when the use of different contract types provide an advantage to the PM.

3. Model for Software Corrective Action Programs

Develop a model for software corrective action to be used by DOD program offices experiencing software developmental problems. The focus should be on a model or series of models that would address the common requirements of any system needing corrective action. The study may include discussion of when a corrective action plan should be initiated and what type programs may need a special model. Additionally, the research would have to determine how the model would assist both the Government and the contractor.

4. EPLRS Follow-on Development

Examine the software challenges in the product improvement phase of the EPLRS program that followed PSV. Hughes learned many lessons from the PSV program in terms of software development and testing. Many of these lessons learned carried over into the next phase of the program. The research could examine how the PSV program influenced follow-on testing of software and system equipment. The team concept has remained in Hughes program offices. This could also be studied for its impact on the follow-on phase. An overall assessment of improved effectiveness in the EPLRS program from the PSV software lessons learned could be the focus of the study.

5. More Studies in Management of Software Corrective Action

Examine the history of other programs that have faced challenges in correcting software problems. The scope of this thesis was limited to the EPLRS program. Studying the success or inability of other programs to effectively manage corrective software action can expand the understanding of the topic. Areas of interest would include other techniques used to effectively manage the corrective action and programs that managed corrective action without breaching baseline or restructuring the program. Also, programs that did not effectively manage corrective action may offer insight to the motivations and situation that cause that to happen.

F. A FINAL THOUGHT

A barrier to the effective use of this case as a model is the pressure to succeed in the DOD procurement environment. Programs that have breached their baseline, like EPLRS, customarily restructure themselves and design focused corrective action plans. Programs that have not breached baseline but are still experiencing serious software deficiencies are driven by different pressures. The desire to maintain low visibility and meet originally planned program schedules may motivate a program manager to try and correct serious software deficiencies in conjunction with current program actions. This may work in certain situations. But the effect of adding corrective action tasks to those of a development process already in progress is to reduce the effectiveness of both actions and only add additional complexity.

The DOD procurement environment must be adapted to encourage the most effective approach to correcting software problems. The pressures of the procurement system that motivate PMs to defer the proper corrective action of software problems delay the maturity of the software and the system itself. This often results in operational test problems, delays in fielding, and significant cost and schedule overruns. When its execution is viable, timely and focused corrective action can avoid these problems.

SECTION H. Special Provisions

H.307 Conduct of the proposed correction of technical test deficiencies is contingent on availability of a full complement of Government furnished test instrumentation (Hardware/Software) and data reduction software at the Contractor's facility for the duration of the PSV activities. This must include capability to build/modify scenarios at Fullerton. Each item must be accompanied by installation, usage, and version description documentation, as applicable. If this documentation does not exist for an item, then on-site support at Fullerton from knowledgeable personnel shall be provided by the Government during the PSV activities.

H.308 Any scenarios, test instrumentation, and/or data reduction software shall be under configuration management disciplines equivalent to the EPLRS Development Program during the PSV activities at Fullerton. Any changes to these items subsequent to PSV activities are subject to prior approval by the DA.

H.309 Technical representatives from both EPG and COMARCO shall be provided at the Contractor's facility to support instrumentation and data reduction for the duration of the PSV activities.

H.310 All EPLRS equipment transferred from Ft. Huachuca to Fullerton shall remain in Government custody for duration of PSV activities at the Contractor's facility. Authorization is granted on a case by case basis by the Government RTR to permit installation of temporary modifications in the Government Furnished Property for purposed of test evaluation. System baseline will be established prior to formal field demo.

H.311 The NCS shelter may be shipped without nameplate and nomenclature since new nomenclature has not yet been assigned. Nomenclature will be assigned during P3I, Phase C, if awarded.

SECTION H. Special Provisions

H.312 The drawing delivery is limited to the new drawings prepared for the PSV program. The delivery shall be microfilm aperture cards. Diazo Type 2 Class 2 made from Hughes microfilm CDRL 03AT developed under DAAB-82-J096 contract.

H.313 Formal regression/verification of design modification changes will be demonstrated at the system level in lieu of individual regression tests for each change.

H.314 The contractor shall be authorized by the RTR on a case by case basis to utilize EPLRS P3I assets (if Phase C awarded) to replace PSV GFP failures when PLRS/GFP (DAAB07-83-C-J031) assets are exhausted.

H.315 Any training material that was revised by IMMI shall be furnished to the Contractor by the Government NLT 30 days after execution of the syscon training option.

H.316 Redline technical manuals will be used to support course preparation. Draft manuals reflecting PSV configuration changes will be utilized during course conduct.

H.317 At minimum, one student station will be utilized for spare parts to support hardware maintenance of the trainer.

H.318 Any failed Megatek or Sun Microsystems circuit cards will be replaced with existing trainer assets.

H.319 During actual conduct of training, only hardware maintenance support of the trainer will be provided. However, any software or exercise failures that are determined critical by the Government to training will be processed immediately.

H.320 The latest date that the option may be exercised by is Oct 1989.

H.321 The IC Demo Report generated by the Contractor is due within 30 days after receipt of Government reduced data from the IC Demo. If IC Demo Report is submitted late to the Government, the PCO may reduce progress payments on the PSV program.

SECTION H. Special Provisions

H.322 If the Contractor fails the IC Demo, the PCO may stop the PSV program progress payments. The Contractor shall analyze the failures and make corrections as required to pass the IC Demo to the Government's satisfaction, at no change to the firm-fixed price.

H.323 If the Contractor fails the PSV Field Demo, the Contractor shall analyze the failures and make corrections as required to pass the PSV Field Demo to the Government's satisfaction, at no change to the firm-fixed price.

H.324 The Contractor must successfully pass, as evidenced by written approval form the PCO, the Inter/Intra Community (IC) Needline Demonstration in order to obtain authorization to implement First Article Test (FAT) requirements under Phase C of this contract if awarded.

H.325 The Contractor must successfully pass, as evidenced by written approval of test report by PCO, the PSV Demonstration in order for Option 1 of Phase C (if awarded) to be exercised. Failure to pass the PSV Demonstration will delay authorization of Option I, but will not be cause for any additional cost or delivery schedule delay when Option I of Phase C is exercised.

H.326 The PSV equipment must successfully pass the Technical Test (Phase 3) as evidenced by written approval by the PCO, in order for Option 2 of Phase C (if awarded) to be exercised. Failure to pass the Technical Test will delay authorization of Option 2, but will not be cause for any additional cost or delivery schedule delay when Option 2 of Phase C is exercised.

APPENDIX B. LIST OF ACRONYMS

ACAT	Acquisition Category
AFATDS	Advanced Field Artillery Tactical Data System
AMSAA	Army Materiel System Analysis Activity
ASARC	Army System Acquisition Review Council
ASAS	All Source Analysis System
ATCCS	Army Tactical Command and Control System
ATF	Advanced Tactical Fighter
bps	bits per second
C/SCSC	Cost/Schedule Control System Criteria
C3I	Command, Control, Communication, and Intelligence
CECOM	Communication-Electronic Command
CFO	California Field Office
COMM	Communication
CSE	Center for Software Engineering
D.A.	Department of the Army
DOD	Department of Defense
DUSA (OR)	Deputy Under Secretary of the Army (Operations Research)
ECM	Electronic Counter Measure
EMD	Engineering and Manufacturing Development

EPG	Electronic Proving Ground
EPLRS	Enhanced Position Location Reporting System
EPUU	Enhanced PLRS User Unit
FAADC2I	Forward Area Air Defense Command, Control, and Intelligence
FAT	First Article Test
FT	Field Test
GAO	General Accounting Office
IC	Inter Community
IOC	Initial Operational Capability
IOT&E	Initial Operational Test and Evaluation
IPR	In Progress Review
IV&V	Independent Verification and Validation
JTIDS	Joint Tactical Information Distribution System
LRIP	Low Rate Initial Production
MANPRINT	Manpower and Personnel Integration
MCCR	Mission Critical Computer Resources
MCS	Maneuver Control System
MCTSSA	Marine Corp Tactical Software Support Agency
NCS	Net Control Station
NCS-E	Net Control Station - EPLRS
OTAR	Over the Air Rekey
OTEA	Operational Test and Evaluation Agency

PDSS	Post Deployment Software Support
PEO	Program Executive Officer
PJH	PLRS-JTIDS Hybrid
PLRS	Position Location Reporting System
PM	Project/Product Manager
PMO	Program Management Office
PQT-C	Prototype Qualification Test - Contractor
PQT-G	Prototype Qualification Test - Government
PSV	Production System Verification
PTR	Program Trouble Report
RDT&E	Research, Development, Test and Evaluation
SED	Software Engineering Division
SEI	Software Engineering Institute
SLOC	Source Lines of Code
SOW	Statement of Work
TAAF	Test Analyze and Fix
TDMA	Time Division Multiple Access
TECOM	Test and Evaluation Command
TEMP	Test and Evaluation Master Plan
TRADOC	Training and Doctrine Command
TSM	TRADOC System Manager
TT-II	Technical Test II

UHF	Ultra-High Frequency

APPENDIX C. LIST OF PERSONNEL INTERVIEWED

1. Guenther, O., MG, USA, Commander, Communications-Electronics Command, Fort Monmouth, NJ, Interview by mail, February 1994.

2. Fornecker, C., LTC, USA, Defense Systems Management College, Fort Belvoir, VA, Interview, June 1993.

3. Frith, S., LTC, USA, Product Manager, Enhanced Position Location Reporting System, Fort Monmouth, NJ, Interview, November 1993.

4. Emery, L., Deputy Product Manager, Enhanced Position Location Reporting System, Fort Monmouth, NJ, Interview, June 1993.

5. Lynn, S., Software Engineer, Enhanced Position Location Reporting System, Fort Monmouth, NJ, Interview, June 1993.

6. Wickstrom, P., Software Engineer, Marine Corp Tactical Software Support Agency, Camp Pendleton, CA, Interviews, August and September 1993.

7. Reska, R., Systems Engineer, Mitre Corporation, Fullerton, CA, Interviews, September and November 1993.

8. Luisi, G., Systems Engineer, Mitre Corporation, Eatontown, NJ, Interview, November 1993.

9. White, L., EPLRS Program Office, Hughes Aircraft Company, Fullerton, CA, Interview, September 1993.

10. Mullen, F., EPLRS Program Office, Hughes Aircraft Company, Fullerton, CA, Interview, September 1993.

11. Ressler, K., EPLRS Program Office, Hughes Aircraft Company, Fullerton, CA, Interview, September 1993.

LIST OF REFERENCES

1. Fairley, Richard E., *Software Engineering Concepts*, McGraw Hill Book Co., 1985.

2. Defense Systems Management College, *Mission Critical Computer Resources Management Guide,* 1990.

3. Defense Science Board, *Report of the Task Force on Military Software.* September 1987.

4. Department of Defense, *U.S. National Strategy*, 1993.

5. Boehm, Barry, "Software Engineering", *IEEE Transactions on Computers*, Vol. C-25, No.12, December 1976.

6. United States General Accounting Office, GAO/IMTEC-92-62BR, *Embedded Computer Systems, Defense Does Not Know How Much It Spends on Software*, Briefing Report to the Chairman, Subcommittee on Research and Development, Committee on Armed Services, House of Representatives. July 1992.

7. Software Engineering Institute Technical Report CMU/SEI-90-TR-12, Carnegie Mellon University, *National Software Capacity: Near-Term Study*, by J.A.L. Siegel, S. Stewman, S. Konda, P.D. Larkey, and W.G. Wagner, May 1990.

8. United States General Accounting Office, GAO/IMTEC-90-23, *Meeting the Government's Technology Challenge: Results of GAO Symposium*, February 1990.

9. Glass, R. L., "The 'Software Crisis' - Is It a Matter of 'Guts Management'?" in *Software Management*, ed. Donald J.Riefer, 4th ed., IEEE Computer Society Press, 1993.

10. Kitfield, J., "Is Software DOD's Achilles' Heel?", *Military Forum*, July 1989.

11. United States General Accounting Office, GAO/IMTEC-90-34, *DOD Embedded Computers, Better Focus on This Technology Could Benefit Billion Dollar Weapons Programs.* Report to the Chairman, Legislation and National Security Subcommittee, Committee on Government Operations, House of Representatives. April 1990.

12. United States General Accounting Office, GAO/IMTEC-93-13 *Mission Critical Systems, Defense Attempting to Address Major Challenges*, Report to the

Chairman, Subcommittee on Research and Development, Committee on Armed Services, House of Representatives. November, 1992.

13. W.S. Humphrey, D.H. Kitson, and T.C. Kasse, "The State of Software Engineering Practice: A Preliminary Report", in *Software Management*, ed. Donald J.Riefer, 4th ed., IEEE Computer Society Press, 1993.

14. Software Engineering Institute, *Maturity Distribution -1991*, SEI Software Process Maturity Paradigm, briefing slide, 1991.

15. United States General Accounting Office, GAO/NSIAD-93-15, *Weapons Acquisition, A Rare Opportunity for Lasting Change,* Report from the Comptroller General of the United States, December 1992.

16. United States General Accounting Office, GAO/IMTEC-93-2, *Embedded Computer Systems, Status of C-17 Software*, Testimony Before the Subcommittee on Legislation and National Security, Committee on Government Operations, House of Representatives, 18 March 1993.

17. Yin, R.K., *Case Study Research Design and Methods*, Sage Publications, 1976.

18. Miles, M.B. and Huberman, A.M., *Qualitative Data Analysis: A Source Book of New Methods*, Sage Publications, 1984.

19. Ross, Jimmy D., General, "Winning the Information War," *Army*, v. 44, no. 2, pp. 26-32, February 1994.

20. The Army Data Distribution System Program Office, *Test and Evaluation Master Plan for Enhanced Position Location Reporting System (EPLRS)*, 7 October 1992.

21. Telephone conversation between L. White, Hughes Aircraft Company, and the author, 1 March 1994.

22. Brown, M. E., and Reska, R. S., "Lessons Learned During Testing of the Enhanced Position Location Reporting System (EPLRS)," Hughes Aircraft Company and the Mitre Corporation, November 1991.

23. Hughes Aircraft Company, *EPLRS/PLRS Program Review*, Briefing slide, 20 November 1987.

24. Enhanced Position Location Reporting System Program Office, *Statement of Work, Completion of EPLRS Phase 5*, 16 May 1988.

25. Electronic Technology Test Directorate, U.S. Army Test and Evaluation Command, *Summary of Observations, Phase I of the Technical Test of Enhanced Position Location Reporting System (EPLRS)*, February, 1989.

26. TRADOC Systems Manager, Army Data Distribution System, *Memorandum for Project Manager, Position Location Reporting System/Tactical Information Distribution System; Subject: User Requirements and Thresholds for Future EPLRS Development and Testing*, 5 July 1989.

27. Office of the Under Secretary of the Army (Operations Research), *Report of the Program Deviation Report Review Panel for Enhanced Position Location Reporting System (EPLRS)*, 21 July 1989.

28. Program Manager, Army Data Distribution System, *Memorandum for Distribution, Subject: Notional EPLRS Restructuring*, 6 June 1989.

29. Hughes Aircraft Company, *EPLRS; Correction of technical Test Deficiencies*, Briefing slide, 14 July 1989.

30. Hughes Aircraft Company, *Correction of Technical Test Deficiencies; Production System Verification (PSV) Program*, Briefing slide, 3 March 1990.

31. Interview between Frank Mullen, Hughes Aircraft Company, and the author, 23 September, 1993.

32. U.S. Army Communications and Electronics Command, Center for Software Engineering, *Software Engineering Assessment of EPLRS*, Briefing slide, 13 July 1989.

33. Enhanced Position Location Reporting System Program Office, *Modification No. P00100 (Production System Verification), to Contract No. DAAB07-83-C-J031 (EPLRS P3I)*, 1 September 1989.

34. Office of the Deputy Under Secretary of the Army (Operations Research), *Memorandum for Army Acquisition Executive; Subject: Addendum to the Program Deviation Report Review Panel for Enhanced Position Location Reporting System (EPLRS)*, 22 May 1990.

35. Interview by mail between O. Guenther, Major General, USA, Communications-Electronics Command, Fort Monmouth, NJ, and the author, 27 February 1994.

36. Defense Systems Management College, *Test and Evaluation Management Guide*, 1988.

37. Interview between C. Fornecker, Lieutenant Colonel, USA, Defense Systems Management College, Fort Belvoir, Virginia, and the author, 23 June, 1993.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey CA 93943-5002	2
3.	Defense Logistics Studies Information Exchange U.S. Army Logistics Management Center Fort Lee, Virginia 23801-6043	1
4.	OASA (RDA) ATTN: SARD-ZAC 103 Army Pentagon Washington, D.C. 20310-0103	1
5.	Professor David V. Lamm, Code SM/Lt Department of Systems Management Naval Postgraduate School Monterey, California 93943-5000	2
6.	Professor Martin McCaffrey, Code SM/Mf Department of Systems Management Naval Postgraduate School Monterey, California 93943-5000	6
7.	Professor Tarek Abdel-Hamid, Code SM/Ah Department of Systems Management Naval Postgraduate School Monterey, California 93943-5000	1

8. MG Guenther 1
Commander, U.S. Army CECOM
Fort Monmouth, NJ 07703-5000

9. Ray Reska 1
Mitre Corporation, c/o Hughes Aircraft Co.
Bldg. 676-D319
1901 Malvern
Fullerton, California 92633

10. Len White 1
M/S FU/676/C345
PO Box 3310
Fullerton, California 92634

11. LTC Steve Frith 1
Product Manager, Enhanced Position Location Reporting System
Bldg 2525, PM ADDS
Fort Monmouth, New Jersey 07703

12. Leo Emery 1
Deputy Product Manager, Enhanced Position Location Reporting System
Bldg 2525, PM ADDS
Fort Monmouth, New Jersey 07703

13. Paul Wickstrom 1
USMC-MCTSSA
ATTN: CN-06C
Camp Pendleton, California 92055

14. LTC (P) Hamilton, Code SM/Ha 1
Department of Systems Management
Naval Postgraduate School
Monterey, CA 93943-5000

15. CPT Terrence Cummings 2
PO Box 1726
Ft. Benning, Georgia 31905-0326